1-1-2018

Mixture Models With Grouping Structure: Retail Analytics Applications

Haidar Almohri
Wayne State University,

MIXTURE MODELS WITH GROUPING STRUCTURE: RETAIL ANALYTICS APLICATIONS

by

HAIDAR ALMOHRI

DISSERTATION

Submitted to the Graduate School

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

2018

MAJOR: INDUSTRIAL ENGINEERING

Approved By:

_____

Advisor Date

_____

_____

_____

DEDICATION

To my lovely daughter Nour.

ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Ratna Babu Chinnam for introducing me to this fascinating research topic and for his guidance, technical advise, and support throughout this research. I am also grateful to my committee members, Dr. Alper Murat, Dr. Evrim Dalkiran for their time and interest in this thesis and for providing helpful comments for my research. Special thanks go to Dr. Arash Ali Amini for providing valuable suggestions to improve this work. I also thank my family and friends for their continuous support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AIC Akaike Information Criteria

BIC Bayesian Information Criterion

CL Competitive Learning

CV Cross Validation

EM Expectation Maximization

FMM Finite Mixture Models

FMR Finite Mixture of Regressnion

GLM Generalized Linear Models

GMR Group Mixture of Regressions

KPI Key Performance Indicators

LASSO Least Absolute Shrinkage and Selection Operator

MAP Maximum a Posterior

MOO Multi-objective Optimization

NMI Normalized Mutual Information

OEM Original Equipment Manufacturers

RIS Retail Information Systems

RMSE Root Mean Squared Error

SNR Signal to Noise Ratio

SSE Sum of Squared Errors

SVM Support Vector Machine

CHAPTER 1: INTRODUCTION

1.1 Background

Increasing global competition combined with product proliferation, dropping customer loyalties, and shrinking product life-cycles is changing the environment facing most companies today. None of the industries seem to be immune and retail and franchise sectors seem to be particularly hurting. The number of retailers filing for bankruptcy protection in the U.S. is headed toward its highest annual tally since the Great Recession in the 1920s [Gustafson, 2017]. While the reasons for bankruptcies and difficulties are several, many companies within these sectors are lacking comprehensive and effective performance management systems [Yu and Ramanathan, 2009]. These sectors have no doubt seen tremendous efficiencies from employing in-store technologies (e.g., scanning systems, enhanced point-of-sale systems, self-service lines) and information technology to drive upstream operations (e.g., warehousing, logistics, and manufacturing) [King et al., 2004]. While these technologies and the associated Retail Information Systems (RIS) are effective in helping to manage core store activities (like inventory control and logistics), these chains also need analytics platforms for performance management [Rigby, 2011, Nash et al., 2013]. The term analytics refers to "the process of developing actionable insights through problem definition and the application of statistical models and analysis against existing and/or simulated future data" [Cooper et al., 2012]. Retail analytics "is any process/information that allows retailers to make smarter decisions and manage their businesses more effectively" [dic, 2017].

With the increasing availability of data and technologies to store and process the data, business analytics has been experiencing growing attention among the researchers and is transforming the way businesses operate in many sectors [Sun et al., 2017, Henke et al., 2016]. In the context of retail and franchise networks, stores need customized analytical guidance to improve profitability and sales of individual stores based on their specific location, demographics, and environment. Executives are unanimous in voicing their concerns over the lack of methods to assess store specific issues and derive equally specific insights [Bucklin and Gupta, 1999]. Such guidance is key to important strategic management decisions, including evaluation, promotion and development. Strategic resource-allocation decisions, such as advertising budgets, store expansions/closings, are also based on management's understanding of store performance drivers [Thomas et al., 1998].

[Parsons, 1994] makes the distinction between store efficiency and effectiveness. Efficiency refers to the relationship between inputs and outputs, while effectiveness focuses on outputs relative to a particular objective. Productivity is the combination of efficiency and effectiveness and is the focus of this study. [Thomas et al., 1998] note that productivity studies demand several careful considerations. First, relevant individual store differences must be considered within the platform to take into account advantages and disadvantages of particular stores, e.g., location, competitive intensity [Kamakura and Ratchfrord, 1996]. Second, development is much more effective when specific practices can be observed and transferred to other stores. Effective practices should be identified, described, and used as benchmarks for less efficient stores. This is a key focus of this study and the proposed method attempts to attribute (through modeling) variation in performance across a

set of stores to the way the stores are managed (in particular, how the Key Performance Indicators (KPI)s are managed using resources). Third, a distinction must be made between resources under the control of store vs. those they have little or no influence over (e.g., local land/rent costs). In our proposed approach, we emphasis deriving recommendations around "actionable" KPIs (e.g., increasing advertising budget) vs. not so actionable KPIs (e.g., changing store location or reducing overhead). Fourth, more than one outcome usually needs to be considered because stores are responsible for multiple and sometimes conflicting performance measures (e.g., dealership sales might be more important to an automotive manufacturer vs. profits to the owner's of the dealership). This is also addressed in this study through the use of multi-objective optimization methods for deriving recommendations.

Numerous methods have been proposed for evaluating retail efficiency of individual stores [Balakrishnan et al., 1994, Kamakura and Ratchfrord, 1996]. Past research has shown that store performance is influenced by trade area demographic factors [Ingene and Lusch, 1980], level of competition [Craig et al., 1984], retail atmospherics [Jain and Mahajan, 1979], and promotions [Walters and MacKenzie, 1988]. Others have researched the effect of internal retail environment including level of service and extended store hours [Kumar and Karande, 2000] as well as overlapping trading areas [Pauler et al., 2009]. As for methods, Data Envelopment Analysis (DEA) has been widely used for benchmarking performance of retail stores [Kamakura and Ratchfrord, 1996, Donthu and Yoo, 1998, Vyt, 2008]. Another common approach that is adopted by economists for evaluating the efficiency of retail stores is using translog (Transcendental Logarithmic) cost function, a

second-order approximation to a cost function that can be used to model how a firm combines inputs to produce outputs [Caves et al., 1982, Kamakura and Ratchford, 1996].

The focus in this dissertation is to facilitate improvement in the performance of individual stores by relying on a data-driven approach to internal benchmarking. In particular, the goal is to identify factors driving automotive dealership performance in comparison with "similar" dealerships and relying on optimization to derive tailored recommendations. The problem was brought to our attention by a global leader in providing automotive dealership location and network analysis to many automotive Original Equipment Manufacturers (OEM)s.

In the automotive industry, dealer efficiency and effectiveness are key factors for obtaining and maintaining competitiveness for OEMs. This is just as critical for the well-being and durability of dealers, for most dealerships tend to be franchises (in the U.S. and much of the world) that have a contract with an automotive OEM that allows them to sell its products. It is critically important to establish analytics platforms for assessing the productivity of the dealer network that is not only useful for the OEM but also provide customized guidance to individual dealerships. Automotive OEMs usually assess dealership performance according to market share and plan incentive systems by assigning annual sales targets to each dealership. However, performance assessment based on simplistic comparison between the dealership and national or state average market share can not only lead to ineffective sales targets but could also compromise the productivity of the dealer and the competitiveness of the OEM [Biondi et al., 2013]. As noted by Biondi and co-authors, this kind of assessment does not take into account either the availability or the

utilization of resources. Dealership A may be more efficient than dealership B according to the market share method, although A can obtain a higher output than B (i.e., sell more vehicles) merely because A has a more consolidated presence in the territory (i.e., has had a sales mandate for longer) and/or is located in a more favorable geographical market (e.g., where the brand enjoys more loyalty). Therefore, a more objective modeling and analysis platform is necessary for evaluating and improving the performance of dealerships.

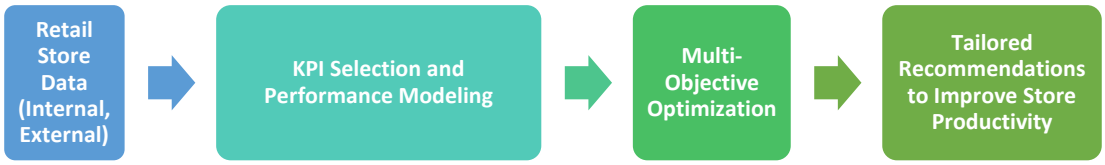


Figure 1.1: Retail Performance Analytics Platform

1.2 Motivation

The company wanted to develop a dealership performance management analytics platform to analyze monthly operations and financial data (including information on sales staffing levels/tenure, product assortment/mix, dealer services (e.g., financing, trade-ins, collision repair), advertising budgets/mix, service bays/technicians etc.) from thousands of dealers in the U.S. to understand factors that can jointly improve profitability for the dealership while also improving vehicle sales to satisfy OEM requirements. In the absence of objective data-driven analytics platforms, dealerships mostly rely on experienced consultants and ad hoc guidance from field personnel. We propose effective model-based methods for clustering the stores into similar groups for benchmarking. Figure 1.1 illustrates our overall approach.

1.3 Contribution

Our contributions are as follows: 1) We propose an objective method for segmenting retail stores (in particular, automotive dealerships) using a model-based clustering technique that accounts for similarity in store performance dynamics, 2) We propose an effective Finite Mixture of Regressions (FMR) technique based on competitive learning for carrying out the model-based clustering and modeling store performance, 3) While the competitive learning technique proved to produce good results for the dealership case study, we also provide an exact Expectation Maximization method to this problem through what we called Group Mixture of Regressions (GMR), and 4) We propose an optimization framework to derive tailored recommendations for individual stores within store clusters that jointly improves profitability for the store while also improving sales to satisfy OEM/franchiser requirements. We illustrate the methods using synthetic experiments and a real-world dataset from a leading global OEM.

1.4 Outline

The rest of the dissertation is organized as follows: Chapter 2 describes the proposed mixture model with completive learning *MMCL* for the problem of finite mixture of regressions (FMR) under group structure constraints, Chapter 3 provides solution to the same problem using Expectation Maximization (EM), Chapter 4 describes a method for deriving tailored recommendations using cluster specific component models using multi-objective optimization, Chapter 5 describes results from a dealership case study, and Chapter 6 offers some concluding remarks and directions for future research.

CHAPTER 2: MIXTURE MODELS WITH COMPETITIVE LEARNING

2.1 Introduction

As mentioned in the Introduction chapter, the motivation for this research stems from assisting a supplier of strategic and operational planning solutions for the automotive sector who is a leader in providing dealership location and network analysis to many automotive OEMs. Beside external factors (e.g., demographics and proximity of competitive dealers), we believe that internal factors (e.g., product carried, quality of workforce, operational processes) that are under more control of the management are also at play. Given our desire to model the performance of dealerships as a function of the KPIs, we need to explicitly handle the presence of dependent variables (e.g., standardized dealership sales and profits calculated relative to averages).

Therefore, we provide a solution for clustering the dealerships into sub-groups by considering both internal external factors. The general idea, which can be employed in clustering any groups of observations (e.g. stores, products, etc.) is presented in this chapter.

2.2 FMM: Background

Clustering is the process of finding subsets of a dataset based on "similarity", where members of the subsets are similar and members across subsets are dissimilar [Guha and Mishra, 2016]. There are several algorithms that have been proposed for the clustering problem [Xu and Wunsch, 2005]. Traditional clustering methods for the most part are heuristic techniques derived from empirical methods and have difficulty taking into ac-

count the characteristics of clusters (shapes, proportions etc.). Finite mixture models have attracted much attention in recent years for clustering. [McLachlan and Basford, 1988] were the first to highlight the usefulness of mixture models as a way of providing an effective clustering of various datasets under a variety of experimental designs. They offer considerable flexibility and permit certain classical criteria for vigorous analysis and have been widely used for market segmentation and similar studies [Green et al., 1976, Gupta and Chintagunta, 1994, Jedidi et al., 1997, Andrews et al., 2011, Wedel and Desarbo, 2002, Sarstedt, 2008, Tuma and Decker, 2013a].

One of the challenges in modeling certain populations is that the observations might be drawn from different distributions/processes underlying the overall population. In such cases, a "single" model may fail to efficiently represent the sample data and therefore the accuracy and reliability of the model might suffer. This problem has been identified more than hundred years ago [Newcomb, 1886, Pearson, 1894] and "mixture" models were introduced in order to better account for the unobserved heterogeneity in the population. Since those early days, a lot of effort has gone into developing new methodologies and to further improve the modeling. In recent years, due to increasing availability and diversity of data, the topic has experienced an increasing attention by researchers. Mixture models have been successfully employed in a variety of diverse applications such as speech recognition [Reynolds et al., 1995], image retrieval [Permuter et al., 2003], term structure modeling [Lemke, 2006], biometric verification [Stylianou et al., 2005], and market segmentation [Tuma and Decker, 2013b].

In mixture models, "components" are introduced into the mixture model to allow for

greater flexibility in modeling a heterogeneous population that is apparently unable to be modeled by a single model. The hope is that this form of clustering would allow for more effective modeling and comparison of stores within individual clusters and for recognition/extraction of effective practices to be used as benchmarks for less efficient stores.

In mixture models, it is assumed that the observations are generated according to several probability distributions (a.k.a. components) with certain parameters. Data points in each distribution are assumed to form a cluster. The general framework of FMMs is of the following form:

$$f(x) = \sum_{k=1}^{K} \alpha_k f(x_k; \theta_k) \tag{2.1}$$

where k is the number of mixture components, α_k is the mixing weights ($\alpha_k > 0$ and $\sum_k \alpha_k = 1$), θ_k is the set of parameters for the k^{th} component, and $f(x_k; \theta_k)$ is the distribution of the k^{th} component. Each component k is assumed to come from a unique $f(x_k; \theta_k)$, which is some probability distribution, with probability α_k that an observation comes from component k. In the case of mixture of Gaussian distributions, $f(x_k; \theta_k) \sim \mathcal{N}(\mu, \Sigma)$, where μ and Σ denote the mean and covariance matrix for each component distribution, respectively.

Among the family of mixture models, the finite mixture of regression (FMR) models have been particularly popular in various fields and applications [Bierbrauer et al., 2004, Andrews and Currim, 2003, Bar-Shalom, 1978], mainly because of the advantages of linear models such as simplicity, interpretability, and scientific acceptance. In FMR, it is assumed

that the distribution of the data can be represented using a convex combination of a finite (K) number of linear regression models. Equivalently, each observation belongs to one the K classes, and given the class membership, it follows the regression model associated with that class. The difficulty is that the class memberships are not known in advance.

Assuming that the dataset consists of n observations $(y_i, x_i), i = 1, \ldots n$, let y_i denote the value of response variable for the ith observation, and x_i the corresponding $p \times 1$ vector of independent variables (for brevity, we exclude the intercept from the notation). Let $y = (y_1, y_2, \ldots, y_n) \in \mathbb{R}^n$ be the response vector, and $X = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^{n \times p}$ the design matrix. Then we can write:

$$y_i = \sum_{k=1}^{K} \alpha_k \phi\big(\frac{y_i - x_i\beta_k}{2\sigma_k^2}\big) \qquad k = 1, ..., K \tag{2.2}$$

where ϕ is the normal density with mean $x_i\beta_k$ and variance σ_k^2, β_k is the regression coefficient of the kth component, K is the number of linear regression models (i.e. components), and α_k is the mixture probability (the proportion of kth component with respect to the total population; $\sum_{k=1}^{K} \alpha_k = 1$). Here, we assume that K is known in advance. The ultimate objective is to estimate the parameters of the mixture model. In the case of FMR, the parameters to be estimated are: $\Theta = (\pi_1, \ldots, \pi_K, \beta_1, \ldots, \beta_k, \sigma_1, \ldots, \sigma_k)$.

[Quandt and Ramsey, 1978] proposed a method of moment algorithm to estimate the parameters of FMR with the presence of a dependent variable. In this setup, it is assumed that the data points ($X \in \mathbb{R}^p$) have an associated dependent variable ($Y \in \mathbb{R}$), and the relation between X and Y is linear: $Y = X\beta$. Therefore, using FMR for modeling and

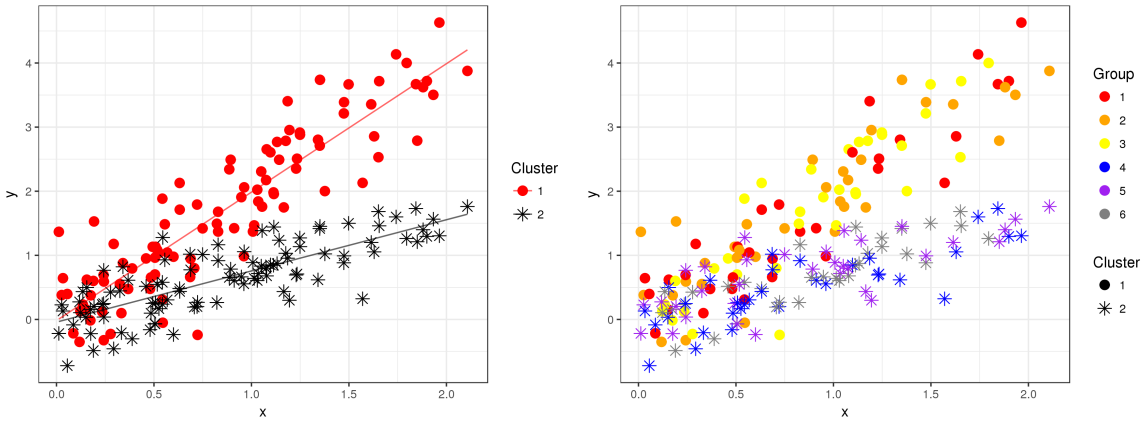
clustering the dealerships would be beneficial.

[Wedel and DeSarbo, 1995] proposed a maximum likelihood solution to extend FMR to Generalized Linear Models (GLM). Since then, several extension for mixture of GLMs has been proposed in the literature (see [Grün and Leisch, 2008], [Raim et al., 2017], [Raim et al., 2017], [Hannah et al., 2011]).

2.2.1 FMR with Group Structure

Under regular FMR, the outcome is a mixture model that provides class membership for each observation of the dataset along with probability (proportion) for each component and the parameters of the model. This results in (soft) clustering the observations into K clusters, assuming K components are employed. In some applications however, instead of individual observations, groups of observations need to be clustered or associated with the same component. For example, if FMR is being employed to model data from a retail chain, it might be necessary to associate all observations stemming from any single store to the same component. This problem is similar to what is known as "clustering with must-link constraint", which is introduced by Wagstaff and Cardie (2000) in the literature [Wagstaff et al., 2001]. The main idea is to utilize experts domain knowledge prior to clustering process in order to obtain desired properties from the clustering solution. Figure 2.1 illustrates the concept. The data points are synthetically generated using two components: $y_1 = \frac{1}{2}x + \epsilon_1$ and $y_2 = \frac{3}{4}x + \epsilon_2$, where $x \sim \mathcal{N}(0, 1)$, $\epsilon_1 \sim \mathcal{N}(0, 0.5)$, and $\epsilon_2 \sim \mathcal{N}(0, 0.3)$. Figure 2.1a shows the linear relationship between the two groups (y_1 and y_2), without any grouping (must-link) structure. In figure 2.1b, the data points are linked to create six (6) groups (groups 1-3 belong to y_1 and groups 4-6 belong to y_2). The data points with the

same color refer to the same group. The desired outcome is that all the data points in the same group end up having the same class membership. See [Basu, 2009] for an extensive review of constrained clustering algorithms and applications.



(a) (b)

Figure 2.1: FMR with "group" structure constraints: (a) Synthetic, two component FMR without any constraint. (b) The same data points being divided into six groups where each group has to retain its data points.

To the best of our knowledge, all the existing algorithms have solved the problem of clustering with group structure in unsupervised/semi-supervised settings, meaning that the observations lack (or partially lack) the dependent variable. In other words, we could not find any work that addresses FMR with grouping structure. Also, most of the solutions that are provided to solve this problem in the literature are parametric, where there are assumptions on the distribution of the data, and the solutions estimate the parameters of the distribution. In this work, we provide a non-parametric solution to FMR with grouping structure. The advantage of our proposed algorithm is that it can be employed with any prediction technique (i.e. Support Vector Machine (SVM) regression, random forest, neural networks, etc.), a significant advantage when modeling data with non-linear relationship.

2.3 Mixture Model with Competitive Learning

In this section, we propose an algorithm that relies on "Competitive Learning (CL)" for FMR with group structure constraints, labeled Mixture Model with Competitive Learning (MMCL). Later sections illustrate how this algorithm can be employed to address the problem of automotive dealership clustering and performance management. We also offer guidance on parameter initialization for *MMCL*.

2.3.1 Competitive Learning

Competitive learning is a form of unsupervised learning originating in the domain of artificial neural networks, in which nodes of the network compete for the right to respond to a subset of the input data [Rumelhart et al., 1988]. A variant of Hebbian learning, standard competitive learning algorithms work by increasing the specialization of each node in the network (typically composed of a single layer of neurons) and is well suited to finding clusters within data. There are three basic elements to the standard competitive learning rule [Haykin et al., 2009]: 1) A set of neurons that are all the same except for some randomly distributed synaptic weights, and which therefore respond differently to a given set of input patterns; 2) A limit imposed on the 'strength' of each neuron, and 3) A mechanism that permits the neurons to compete for the right to respond to a given subset of inputs, such that only one output neuron, is active (i.e. 'on') at a time. Typically, the neuron that wins the competition is called a "winner-take-all" neuron. Accordingly, during the training cycle, the individual neurons of the network learn to specialize on ensembles of similar patterns and in so doing become "feature detectors" for different classes/clusters

of input patterns.

In what follows, we adapt the standard competitive learning algorithm to the more general model setting of FMR.

2.3.2 MMCL Algorithm

Let D be the complete set of data points and M the number of distinct groups within D (e.g., data from each dealer forms an observation group). Define S as the set that holds all the groups: $S = \{s_i\}_{i=1}^{M}$. Each group s_i, with n_i observations, has to retain all its members when assigned to a cluster, forming the group structure constraints. The goal is to assign each s_i to one of K clusters. Ideally, the dataset is partitioned into training and testing datasets, where a subset of the data from each group is stored in the testing dataset for testing and improving model robustness.

Assuming that the number of components K is known, the proposed *MMCL* iterative algorithm starts by randomly selecting K groups (out of M) for initializing the clusters, and fitting one model using observations in each group to get a function $f_i(x;\theta)$ for each $i \in 1, ..., K$. In the event the individual groups are too small to learn the initial model for each cluster component, one can randomly assign multiple groups to each component for initialization. This approach is generic in that the component model can be of any type, e.g. linear regression, Least Absolute Shrinkage and Selection Operator (LASSO), multi-layer perceptron, support vector machine, etc. Next, all the groups $s_i \in S$ are selected one at a time, and are predicted using each of the K models. The group is assigned to one of the K cluster components that produces the best performance; this is "competition" portion of the *MMCL*.

For performance assessment, one can employ several criteria based on the component model, such as, the Sum of Squared Errors (SSE) of prediction, Bayesian Information Criterion (BIC), or Akaike Information Criteria (AIC), on a held out testing dataset. Performance is evaluated on the testing dataset and not the training dataset to reduce the chance of over fitting. In our experiments with linear regression component models, AIC provided robust performance. Founded on information theory, given a collection of models for the data, AIC estimates the quality of each model relative to each of the other models. In doing so, it deals with the trade-off between the goodness of fit of the model and the complexity of the model and is one of the most common model selection procedures that is available in most statistical software packages [Chaurasia and Harel, 2012]. Akaike stated that modeling is not only about finding a model which describes the behavior of the observed data, but its main aim is predicated as a possible good, and the future of the process is under investigation [M., 2014]. The AIC is calculated as:

$$AIC = -2l_f(\hat{\theta}) + 2k \tag{2.3}$$

where $l_f(\hat{\theta})$ is the maximum value of the likelihood function of the model with parameters $\hat{\theta}$. Given that models with minimum AIC are preferred, AIC employs the term $2k$ to penalize complex models with more parameters.

Without loss of generality, in the rest of the manuscript, we assume that AIC is the criterion for cluster component competition within *MMCL*.

After assigning each group to the cluster with the best (i.e., minimum) AIC, the "Overall

AIC" is calculated as the sum of the Cluster AICs: $Overall\ AIC = \sum_{j=1}^{K} Cluster.AIC_j$. In the next iteration, the process is repeated and this time the K component models are learnt using all the aggregated observations (that consist of several groups) within each of K clusters; this is the "learning" portion of *MMCL*. The updated models will compete in the same fashion as described above, to select the cluster members that produce the lowest AIC. This process is repeated until algorithm convergence (e.g., no changes in cluster group memberships between iteration t and $t+1$ and/or the resulting Overall AIC) or if the maximum number of iterations has been reached. This is a common approach to stop searching in most of the meta-heuristic optimization methods [Safe et al., 2004].

Let us denote $\mathcal{M}_j$ as the j^{th} model (that is built using the observations in j^{th} cluster) and $\mathcal{A}_{\mathcal{M}_j,s_i}$ as the AIC value resulting from predicting the response values in group s_i using model $\mathcal{M}_j$. The ultimate goal is to predict a label $c^{(i)}$ for each group s_i. We can write the algorithm in the form of an optimization problem as follows:

$$c^{(i)} = \underset{j}{\operatorname{argmin}}\, \mathcal{A}_{\mathcal{M}_j,s_i} \tag{2.4}$$

It can be seen that *MMCL* is a special version of the famous *k-means* clustering [Steinhaus, 1956], with the distance defined as the AIC of predictive models.

In the case of a linear regression component model, Eqn. 2.4 can be written as:

$$c^{(i)} = \underset{j}{\operatorname{argmin}}\, nlog\left|\frac{RSS_{\mathcal{M}_j,s_i}}{n_i}\right| + 2p \tag{2.5}$$

where p is the number of parameters in the model and $RSS_{\mathcal{M}_j,s_i} = \sum_{l=1}^{n_i} (\hat{y}_{\mathcal{M}_j,s_{il}} - y_{s_{il}})^2$

is the residual sum of squares resulting from predicting response variables in group s_i using model $\mathcal{M}_j$. It is easy to observe that the with a minor modification, any stopping criteria and model selection technique can be used.

The Psuedo code for *MMCL* is provided in Algorithm #1.

Algorithm 1 MMCL for FMR with Group Structure Constraints

procedure – COMPETITIVE LEARNING
Initialize AIC to a large value (e.g., 10^{10}) and ϵ to a small value (e.g., 0.001)
Randomly select K observation groups for initializing each of the cluster component models
repeat
$AIC_{Old} = AIC$
Learn the K component models using observations assigned to the K clusters
for each group $s_i \in S$ **do**
Make predictions for selected group using each of K models and record AIC
Assign s_i to cluster with the least AIC
end for
Calculate $Cluster.AIC_j, \forall j = 1, \ldots, K$
$AIC = \sum_{j=1}^{K} Cluster.AIC_j$
until convergence (i.e., $|\frac{AIC - AIC_{Old}}{AIC_{Old}}| < \epsilon$)
end procedure

2.3.3 Initializing MMCL Parameters

There are three main parameters that need to be selected prior to applying *MMCL*:

1) Number of clusters K: An existing method such as BIC, Calinsky criterion, Gap Statistics, etc. can be used. With the existence of response variables (supervised learning), it is recommended to divide the data to train and test sets and select K that yields the best result.

2) Initializing K clusters: This is a critical step as it effects both the convergence and effectiveness of the algorithm. One of the existing methods such as *k-means++*

[Arthur and Vassilvitskii, 2007] can be adopted to develop an algorithm that wisely selects the initial cluster groups. *k-means++* has two steps for selecting the cluster centroids:

a. Select one center from the data points uniformly at random
b. Compute the distance $\mathcal{D}(x)$ between each data point and the centers that have already been selected
c. Select a new center with probability proportional to $\mathcal{D}(x)^2$
d. Repeat steps (a) and (b) until all K centers are selected
e. Apply *k-means* using the selected points as initial cluster centroids

Inspired by *k-means++*, the following initialization algorithm (labeled *MMCL++*) is developed to select the initial groups. It tries to smartly choose the groups so that the selected groups have maximum dissimilarity. This is achieved by selecting a group $s_j \in S$ at random and predicting all the remaining groups using the model $\mathcal{M}_j$ that is developed by the observations in that group. The quality of prediction for all the remaining groups using $\mathcal{M}_j$ is evaluated, and the group $s_i, i \neq j$ that $\mathcal{M}_j$ has the least power predicting it is identified as the candidate that has the maximum distance with s_i. Again, different criterion such as correlation between s_i and s_j, RSS of $\mathcal{M}_j, s_i$, etc. can be used for this purpose.

3) Parameters of the models θ: With the current version of the algorithm, the parameters of the models can only be optimized using cross validation.

Algorithm 2 *MMCL++ Initialization Algorithm*

repeat
Select one observation group $s_j \in S$ at random
Learn model $\mathcal{M}_j$ using the observations in s_j
Predict remaining groups $s_i, i \neq j$ using $\mathcal{M}_j$ and calculate $\mathcal{A}_{\mathcal{M}_j,s_i}, \forall\, i = \{1,\ldots,M\} \wedge i \neq j$
Select a new group s_i that has $max(\mathcal{A}_{\mathcal{M}_j,s_i}), i = \{1,\ldots,M\} \wedge i \neq j$
until K groups are selected

2.4 MMCL Validation: Synthetic Experiments

To evaluate the effectiveness of the proposed *MMCL* algorithm for FMR with group structure constraints, we employ Monte Carlo simulation experiments.

2.4.1 Experiment Setup

For the synthetic experiments, for ease of exposition, it was decided to use linear regression as the modeling technique. The impact of different parameters on the result is investigated. The experiment is conducted for the case $K = 2$ (number of clusters). Covariates (X) for each cluster are generated by drawing samples from a bivariate Gaussian distribution: $X \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, with zero mean and a diagonal covariance matrix with unit variance.

Table 2.1: Monte Carlo Simulation Parameters (MMCL)

	N	S	Noise Level	d^2
Cluster 1	300	5	(0.5, 1, 2, 4, 6)	(0.2, 0.6, 1.8)
Cluster 2	300	15		

Referring to the Monte Carlo Simulation Parameters outlined in Table 2.1, $N = 300$ is the total number of observations in each cluster and S is the number of groups (blocks) in each cluster. Essentially, there will be 60 observations per group in cluster 1 and 20

observations per group in cluster 2. The response variable for each observation is generated by: $y_i = X_i^{'}\beta + Noise\ level$. The "Noise level" parameter is used to control the amount of noise (uncertainty) added to the response variable y. It can also be seen as the parameter that controls the Signal to Noise Ratio (SNR).

To study the effect of the degree of similarity between βs, the Euclidean distance (d^2) between β_1 and β_2 is calculated to control the level of separation for the two clusters (in the response domain). That is: $d^2 = \left|\left|\beta_1 - \beta_2\right|\right|^2 = \left|\left|\beta_1\right|\right|^2 + \left|\left|\beta_2\right|\right|^2 - 2\left\langle\beta_1, \beta_2\right\rangle = 2(1 - r_{12})$, where $r_{12} = \langle\beta_1, \beta_2\rangle$, assuming that $\left|\left|\beta_1\right|\right|^2 = \left|\left|\beta_2\right|\right|^2 = 1$ (βs have l_2 norm of one). If R=[1 r_{12}; r_{12} 1], and the matrix B is the Cholesky decomposition of R, then the i^{th} row of B is β_i, with square distance d^2 between β_1 and β_2. Obviously, the smaller d^2, the closer the βs, and it is harder to separate the clusters.

2.4.2 Results from MMCL

The Monte Carlo simulations are repeated a 1000 times for each pair of d^2 and Noise level. Normalized Mutual Information (NMI) is used for assessing the clustering accuracy. NMI is a widely used technique in evaluating the clustering result when the true labels are available. The advantage of using NMI is that it is independent of permutation, meaning that the label switching does not affect the NMI score. It is bounded between zero and one. The closer the value to zero, the higher the indication that the cluster assignments are largely independent, while NMI close to one shows substantial agreement between the clusters. An NMI value of zero simply means that the label assignment is purely random. Figure 2.2a shows the average NMI (for 1000 runs) for different levels of noise and d^2, while Figure 2.2c shows the distribution of NMI among 1000 runs.

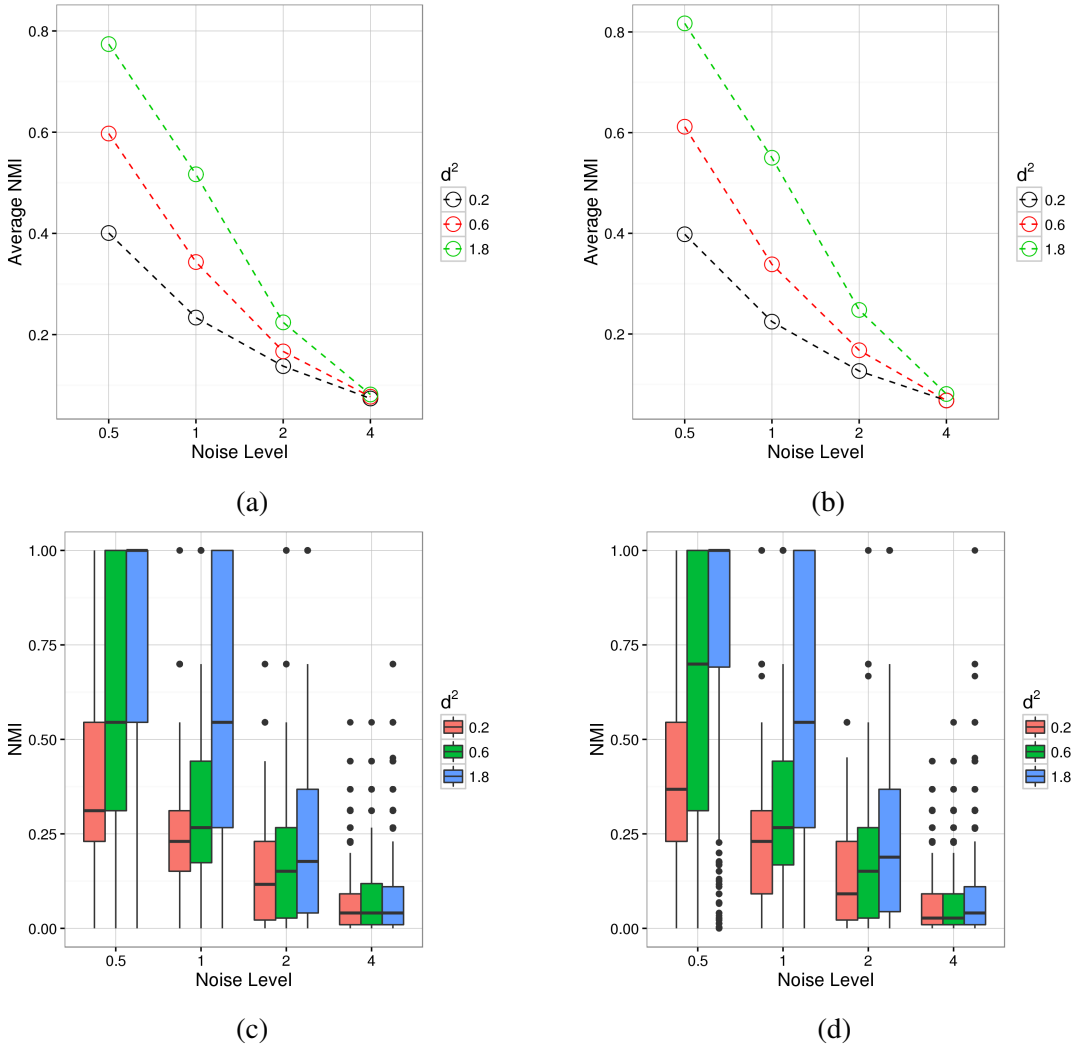
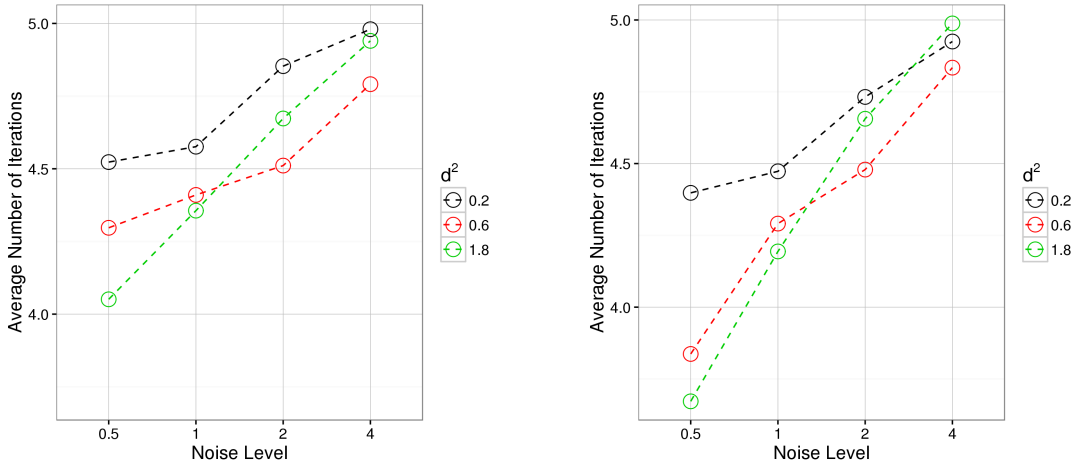


Figure 2.2: MMCL Performance on Synthetic Datasets: (a) Average NMI for different noise levels and d^2 (random initial groups assignment). (b) Average NMI for different noise levels and d^2 using *MMCL++*. (c) Distribution of NMI for different noise levels and d^2 (random initial groups assignment). (d) Distribution of NMI for different noise levels and d^2 using *MMCL++*.



(a) (b)

Figure 2.3: Evaluating *MMCL* Efficiency in Iterations: (a)Average number of iterations for different noise levels and d^2 (random initial groups assignment). (b) Average number of iterations for different noise levels and d^2 using *MMCL++*.

It is evident from Figures 2.2a and 2.2c that *MMCL* is able to achieve a high NMI value (about 0.8) when the noise level is low. However, as expected, as the level of noise added to the response is increased, it significantly affects the accuracy of clustering. With noise level of four, the clustering is almost done in a random fashion. We can also observe the effect of d^2. As mentioned earlier, smaller d^2 means that βs are closer and more similar to each other. Figures 2.2a and 2.2c confirm that as d^2 gets smaller, it is harder to correctly cluster the observations.

Figure 2.3a shows the average number of iterations it took for *MMCL* to converge under different scenarios. It can be seen that on average the algorithm converges within few iterations. Note that the number of iterations is highly affected by the stopping criteria. As mentioned earlier, the stopping criteria is based on the relative change in the overall AIC in two consecutive iterations, i.e. $|\frac{AIC - AIC_{Old}}{AIC_{Old}}| < \epsilon$. In this experiment, ϵ is set to 0.001 and the maximum number of iterations is set to 10.

To observe the impact of careful initial group selection (*MMCL++*), we can compare Figure 2.2a with 2.2b and 2.2c with 2.2d and note that there is a slight improvement in the result when *MMCL++* is utilized, especially when the noise level is low (0.5). Figures 2.3a and 2.3b compare the average number of iterations until convergence between random initial group selection (a) and *MMCL++* (b). The graphs clearly show the effectiveness of *MMCL++* in reducing the number of iterations, especially under low noise level.

2.5 Conclusion

In this study, we proposed a novel model-based clustering algorithm in the context of Mixture Models with grouping structure (must-link constraint). Unlike most of the existing algorithms that are designed to statistically model and estimate the parameters under different probability distributions, our proposed algorithm is non-parametric and can be utilized with any predictive model (i.e. SVM regression, random forest regression, neural networks, etc.) as the underlying regression model. In addition, the proposed method also naturally supports any requirement that seeks to assign groups of observations to individual clusters rather than individual observations. It is designed based on the idea of Competitive Learning where the initial models compete each other for adding the groups to their own cluster. Thus, we called the algorithm Mixture Model with Competitive Learning (MMCL). MMCL is an iterative algorithm that tries to minimize an objective function (i.e. prediction RMSE, prediction R^2, AIC, BIC, etc.), and simultaneously assigns groups of observations to form homogeneous clusters.

The proposed method is validated using synthetic experiments and proved to be effective in recovering the underlying clusters. In Chapter 5, we demonstrate the use of MMCL

for clustering the dealership network.

CHAPTER 3: GROUP MIXTURE OF REGRESSIONS (GMR)

3.1 Introduction

As introduced in Chapter 2 of this dissertation, Mixture Models are used to model complex densities by introducing a mixture of several probability distributions to represent the underlying density. They have been widely used in various applications (see [Böhning, 2000], [McLachlan and Peel, 2004], [Lindsay, 1995]). We also introduced in Chapter 2 the notion of mixture models with group structure (a.k.a mixture models with must-link constraint) and addressed the limitations and lack of methodologies to solve the Finite Mixture of Regressnion (FMR) with group structure. A heuristic method to solve this problem was also proposed in Chapter 2. In this chapter, we demonstrate a parametric approach for solving this problem by modeling and applying Expectation Maximization (EM) to FMR with group structure. We call the algorithm Group Mixture of Regressions (GMR) models.

3.1.1 Estimating the Parameters for Mixture Models

While the parameter estimation in mixture models has been studied mainly from a likelihood point of view [De Veaux, 1989], [Quandt and Ramsey, 1978] used a moment generating function for estimating the parameters. However, maximum likelihood approach using expectation maximization (EM) [Dempster et al., 1977] remains the most widely used technique for estimating the parameters of FMR. EM approach tries to maximize the likelihood in a way that in each iteration, it is guaranteed that the value of likelihood

increases. Other algorithms such as stochastic EM [Celeux and Diebolt, 1985] and classification EM [Celeux and Govaert, 1992] have been introduced as an attempt to improve the performance of the EM algorithm (see [Faria and Soromenho, 2010]). Others have employed Gibbs sampler [Diebolt and Robert, 1994]), and Bayesian approach for estimation [Hurn et al., 2003]. [Chaganty and Liang, 2013] employed low-rank regression with a tensor power method as an alternative to EM algorithm for estimating the parameters.

3.1.2 Grouped Mixture of Regression Models

We assume that the observations belong to R *known groups*, denoted with labels $[R] := \{1, \ldots, R\}$. In each group $r \in [R]$, we observe n_r samples $(y_{ri}, x_{ri}), i = 1, \ldots, n_r$ where $y_{ri} \in \mathbb{R}$ is the response variable and $x_{ri} \in \mathbb{R}^p$ is the vector of covariates or features. We will write x_{rij} to denote the j^{th} feature in the feature vector x_{ri}. For the most part, we will treat x_{ri} as deterministic observations, i.e., we have fixed design regression models.

We assume that there are K latent (unobserved) clusters such that all the observations in group r belong to that cluster. Thus, we can assign a cluster membership variable $z_r \in \{0, 1\}^K$ to each group $r \in [R]$. We will have $z_{rk} = 1$ iff group r belongs to cluster k. With some abuse of notation, we also write $z_r = k$ in place of $z_{rk} = 1$. Given the cluster membership variable z_r, we assume that the group r observations are independent draws from a Gaussian linear regression model with parameters specified by z_r, that is,

$$p(y_{ri} \mid z_r = k) \overset{\text{indept}}{\sim} \mathcal{N}(\beta_k^T x_{ri}, \sigma_k^2), \; i = 1, \ldots, n_r, \tag{3.1}$$

where $\beta_k \in \mathbb{R}^p$ is the coefficient vector the k^{th} regression model and σ_k^2 is the noise variance

for component k. Note that we are assuming that the noise level only depends on the underlying cluster and not on the group. We write $\beta = (\beta_1 \mid \cdots \mid \beta_K) \in \mathbb{R}^{p\times K}$ and $\sigma^2 = (\sigma_1^2, \ldots, \sigma_K^2) \in \mathbb{R}^K$.

As is common in mixture modeling, we assume that z_r follows a multinomial prior with parameter $\pi = (\pi_k)$, that is, $\mathbb{P}(z_r = k) = \pi_k$ for $k \in [K]$, and $z_1, \ldots, z_R$ are drawn independently. The joint distribution of y_r and z_r is then given by:

$$p_\theta(y_r, z_r) = p_\theta(z_r) \prod_{i=1}^{n_r} p_\theta(y_{ri} \,|\, z_r) = \prod_{k=1}^{K} \Big[\pi_k \prod_{i=1}^{n_r} p_\theta(y_{ri} \,|\, z_r = k) \Big]^{z_{rk}} \tag{3.2}$$

where we have let $\theta = (\beta, \pi, \sigma^2)$ collect all the parameters of the model. From (3.1), we have $p_\theta(y_{ri} \,|\, z_r = k) = \phi_{\sigma_k}\big(y_{ri} - \beta_k^T x_{ri}\big)$, where $\phi_\sigma(\cdot)$ is the density of the Gaussian distribution $\mathcal{N}(0, \sigma^2)$. Therefore, the so-called complete likelihood of θ given (z, y) is:

$$L(\theta \,|\, y, z) = p_\theta(y, z) = \prod_{r=1}^{R} p_\theta(y_r, z_r) = \prod_{r=1}^{R} \prod_{k=1}^{K} \Big[\underbrace{\pi_k \prod_{i=1}^{n_r} \phi_{\sigma_k}\big(y_{ri} - \beta_k^T x_{ri}\big)}_{=:\, \gamma_{rk}(\theta)} \Big]^{z_{rk}} \tag{3.3}$$

The parameter $\gamma_{rk}(\theta)$ in (3.3) is proportional (in k) to the posterior probability of z_r given the observation y_r, that is, $p_\theta(z_r = k \,|\, y_r) \propto_k p_\theta(y_r, z_r = k) = \gamma_{rk}(\theta)$. By normalizing $\gamma_{rk}(\theta)$ over k, we obtain the *posterior probability of cluster assignments*:

$$p_\theta(z_r = k \,|\, y_r) = \frac{\gamma_{rk}(\theta)}{\sum_{k'} \gamma_{rk'}(\theta)} =: \tau_{rk}(\theta), \tag{3.4}$$

for any $k \in [K]$ and $r \in [R]$. We note that the overall posterior factorizes over groups, i.e.,

$p_\theta(z \mid y) = \prod_r p_\theta(z_r \mid y_r)$, so it is enough to specify it for each pair z_r and y_r. Thus, $\tau_{rk}(\theta)$ is the posterior probability that group r belongs to cluster k, given all the observations y. These posterior probabilities are key estimation objectives. An estimate $\widehat{\theta} = (\widehat{\beta}, \widehat{\phi}, \widehat{\sigma}^2)$ of θ can be obtained by maximizing (3.3). The classical approach to performing such optimization is by the Expectation Maximization (EM) algorithm, the details of which will be given in Section 3.1.4. Once we have the estimate $\widehat{\theta}$ of the parameters, we can calculate an estimate of the posterior probabilities as $\tau_{rk}(\widehat{\theta})$.

3.1.3 Posterior Prediction with GMR

Now assume that we have new test data point $(y_{r,\text{new}}, x_{r,\text{new}})$ in group r, for which we observe only the feature vector $x_{r,\text{new}}$ and would like to predict $y_{r,\text{new}}$. Let $(y^{\text{train}}, x^{\text{train}})$ denote all the observations used in the training phase. The common link between the training and test data points are the latent variables $z_1, \ldots, z_R$. In other words, since we already have a good estimate of the membership of group r based on the training data (via the posterior (3.4)), we can get a much better prediction of $y_{r,\text{new}}$ than what the prior model suggests. More precisely, we have the following *predictive density* for $y_{r,\text{new}}$ based on y^{train},

$$p_\theta(y_{r,\text{new}} \mid y^{\text{train}}) = \sum_{z_r} p_\theta(y_{r,\text{new}} \mid z_r)\; p_\theta(z_r \mid y^{\text{train}}).$$

Since, $p_\theta(z_r = k \mid y^{\text{train}}) = p_\theta(z_r = k \mid y_r^{\text{train}}) = \tau_{rk}(\theta)$, we obtain the following estimate of the predictive density $p_\theta(z_r = k \mid y^{\text{train}}) = p_\theta(z_r = k \mid y_r^{\text{train}}) = \tau_{rk}(\theta)$. Note that $\widehat{\theta}$ is our estimate of the parameters based on the training data $(y^{\text{train}}, x^{\text{train}})$. In particular, the posterior mean based on (3.4) is $\sum_{k=1}^{K} \tau_{rk}(\widehat{\theta})\, \widehat{\beta}_k^T x_{r,\text{new}}$ which serves as the maximum a

posterior (MAP) prediction for $y_{r,\text{new}}$.

Since the membership group of the new observation is known, we obtain a predictive density for new observations. Thus, we can utilize the group information acquired during training phase. Therefore, we can achieve a better prediction accuracy using the (posterior) latent cluster assignment. This does not happen in usual FMR and is the strength of the proposed GMR.

3.1.4 Parameter Estimation

Let us now derive the EM updates for the model. Recalling (3.3), the complete log-likelihood of the model is $\ell(\theta \mid y, z) = \log p_\theta(y, z) = \sum_{r=1}^{R} \sum_{k=1}^{K} z_{rk} \log \gamma_{rk}(\theta)$ or

$$\ell(\theta \mid y, z) = \log p_\theta(y, z) = \sum_{r=1}^{R} \sum_{k=1}^{K} z_{rk} \Big[\log \pi_k + \sum_{i=1}^{n_r} \log \phi_{\sigma_k}\big(y_{ri} - \beta_k^T x_{ri}\big) \Big]. \tag{3.5}$$

Treating the class latent memberships $\{z_r\}$ as missing data, we perform the EM updates to simultaneously estimate $\{z_r\}$ and θ:

E-Step: We replace (3.5) with its expectation under the approximate posterior of $\{z_r\}$:

$$F(\theta; \widehat{\theta}) := E_{z \sim \tau(\widehat{\theta})}[\ell(\theta \mid y, z)] = \sum_{r=1}^{R} \sum_{k=1}^{K} \tau_{rk}(\widehat{\theta}) \log \gamma_{rk}(\theta) \tag{3.6}$$

using $\mathbb{E}_{z \sim \tau(\widehat{\theta})}[z_{rk}] = \tau_{rk}(\widehat{\theta})$, where $\tau_{rk}(\theta)$ is the posterior given in (3.4).

M-Step: We maximize $F(\theta; \widehat{\theta})$ over θ, giving the update rules for the parameters $\theta = (\beta, \pi, \sigma^2)$.

To derive the update rules, we maximize $F(\theta; \widehat{\theta})$ by a sequential block coordinate as-

Algorithm 3 Grouped mixture of regression (GMR)

1: Compute feature covariances for each group: $\widehat{\Sigma}_r \leftarrow \frac{1}{n_r}\sum_{i=1}^{n_r} x_{ri}x_{ri}^T$
2: Compute feature-response cross-covariances: $\widehat{\rho}_r \leftarrow \frac{1}{n_r}\sum_{i=1}^{n_r} y_{ri}x_{ri}$
3: For any class posterior $\tau = (\tau_{rk})$ define the following weights:

$$\tau_{+k}(\tau) := \sum_r \tau_{rk}, \quad w_{rk}(\tau) := n_r\tau_{rk}, \quad w_{+k}(\tau) := \sum_r w_{rk}, \quad \breve{w}_{rk}(\tau) := \frac{w_{rk}}{w_{+k}}.$$

and the weighted covariances: $\widetilde{\Sigma}_k(\tau) := \sum_{r=1}^R \breve{w}_{rk}\widehat{\Sigma}_r$ and $\widetilde{\rho}_k(\tau) := \sum_{r=1}^R \breve{w}_{rk}\widehat{\rho}_r$.
4: For any parameter $\theta = (\pi, \beta, \sigma^2)$ and class posterior $\tau = (\tau_{rk})$, define the errors:

$$E_{rk}(\beta) := \frac{1}{n_r}\sum_i^{n_r}(y_{ri} - \beta_k^T x_{ri})^2, \quad \overline{E}_k(\beta, \tau) := \sum_r \breve{w}_{rk}(\tau)E_{rk}(\beta)$$

5: **while** not converged **do**
6: Update class frequencies: $\pi_k \leftarrow \tau_{+k}(\tau)/R$, $k \in [K]$
7: Update regression coefficients: $\beta_k \leftarrow \widetilde{\Sigma}_k^{-1}(\tau)\,\widetilde{\rho}_k(\tau)$, $k \in [K]$
8: Update noise variances: $\sigma_k^2 \leftarrow \overline{E}_k(\beta, \tau)$, $k \in [K]$
9: Update class memberships: $\tau_{rk} \leftarrow \tau_{rk}(\theta)$, as given in (3.4), $r \in [R], k \in [K]$
10: **end while**

cent, in each step maximizing over one of the three sets of parameters π, β and σ^2, while fixing the others. The updates are summarized in Algorithm 3. The details can be found in Appendix .1.

3.2 Empirical Analysis

A Monte Carlo simulation study was performed to assess the quality of the GMR algorithm. The results of this study is presented in this section.

3.2.1 Synthetic Data Experiments

To evaluate the effectiveness of the EM algorithm for FMR with group structure constraints, we employ Monte Carlo simulation and modeling experiments.

Table 3.1: Monte Carlo Simulation Parameters

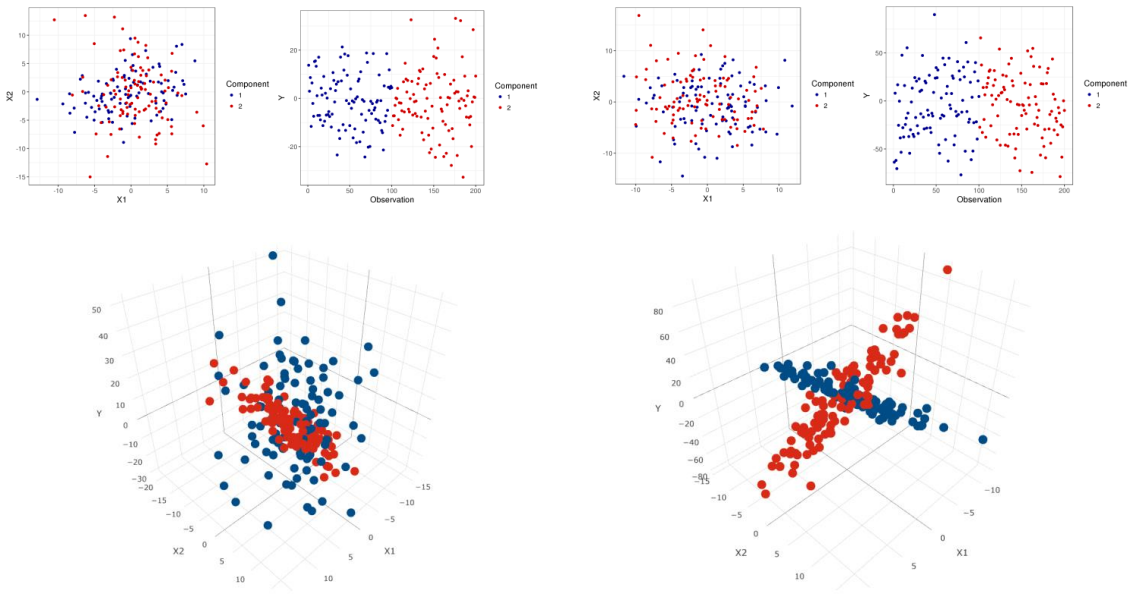
K	d	G	n	Noise Level (σ_k)	β-distance (δ_β)
2	2	10	(100, 200, 400, 800)	(2, 4, 6, 8, 10)	(4, 7, 11)
4	(2, 4)				

Experiment setup. We generate the synthetic data from the GMR model (3.1) with a random design where we generate the feature vectors by drawing each $x_i \sim N(0, \Sigma)$, where Σ is drawn from a normalized Wishart distribution. Recall that K is the number of clusters (or mixture components) and R the number of groups. We will use equal number of observations per group, that is, n_r is the same for all $r = 1, \ldots, R$. Letting $n = \sum_{r=1}^{R} n_r$ be the total number of observations, we will have $n_r = n/R = R/K$ in that case. Let G_k be the number of groups in cluster k. In general, $\sum_{k=1}^{K} G_k = R$; here, we will take all G_k equal so that $G_k = G := R/K$. Thus, it is enough to specify n, G_k, and K. Table 3.1 summarizes various setups used in our simulations. We recall that p is the dimension of the feature vectors x_i (p) and "the noise level" is equivalent to σ_k in (3.1). In each case, the number of groups R and the number of observations per group n_r is determined by the number of clusters K, number of groups in each cluster G_k, and total number of observations n. For example, for $n = 800$, $G_k = 10$, and $K = 2$, we have $R = 20$ and $n_r = 40$.

To study the effect of heterogeneity among regression coefficient vectors $\beta_k, k \in [K]$, we take β_ks to be equi-distant points on a hyper-sphere in $\mathbb{R}^p$ and vary their common distance, denoted as δ_β. More precisely, we will have $\|\beta_k\| = \|\beta_\ell\|$ and $\|\beta_k - \beta_\ell\| = \delta_\beta$ for all $k \neq \ell$. Generating βs this way enables us to compare the estimation errors among different runs of the experiment. The comparison can be carried out across different setups

by normalizing the calculated error, e.g. by the β-distance. Three values of β-distance that are found to be sufficient for our purposes are selected (cf. Table 3.1). Obviously, the smaller the distance, the closer the βs, and it is harder to separate the clusters.

This setup is designed so that the data points are not easily separable in the input and output (X and y) space. The degree of separation is only controlled by β-distance while the noise level (σ_k) controls the uncertainty in relation between X and y. Figure 3.1 shows samples of the generated data for different scenarios.



(a) (b)

Figure 3.1: Sample of the generated data for simulation for the case $p = k = 2$: Covariates X (top left); the response values y (top right); 3d plot for the X and y (bottom): (a) $\delta_\beta = 4$, (b) $\delta_\beta = 12$

Evaluation criteria. The Monte Carlo simulation and modeling experiments are repeated 1000 times for each pair of β-distance and the noise level as well as pairs of d and K. Four criterion are used to benchmark the performance of the algorithm: (1) Normalized mutual

information (NMI) for assessing the clustering accuracy, (2) Average β estimation error, (3) Root Mean Squared Error (RMSE) of prediction to assess the prediction power of the models, and (4) Number of iterations to study the rate of convergence and the speed of the algorithm.

As mentioned earlier in Chapter 2, NMI is an effective metric for evaluating the quality of clustering algorithms when the true labels are available. It is not affected by label switching and penalizes partitions close to random quite aggressively. NMI is bounded between zero and one, where higher values of NMI indicate more agreement between the true and recovered cluster memberships. Note that the output of GMR provides probability of belonging to each groups $1, \ldots, K$. In particular, the resulting τ_{rk} holds soft labeling information. We use Maximum a Posterior (MAP) rule to turn the soft labels to hard labels. For example,if $c^{(r)}$ denotes the assigned (MAP) class membership of group r, then $c^{(r)} = \underset{k}{\text{argmin}} \tau_{rk}$.

"β estimation error" is used as another measure of goodness of fit. We calculate this error by considering both the distance between the true and estimated βs, as well as the miss-classification error. More precisely, to each group r, we can assign two regression coefficient vectors, the estimated one $\widehat{\beta}^{(r)}$, and the true one $\beta^{(r)}$; $\widehat{\beta}^{(r)}$ is equal to $\widehat{\beta}_k$ if we have estimated group r to be in cluster k. Similarly, $\beta^{(r)}$ is equal to β_k if group r is in true cluster k. We can define the average β estimation error as:

$$\text{avg err}_\beta := \frac{1}{R} \sum_{r=1}^{R} \|\widehat{\beta}^{(r)} - \beta^{(r)}\|^2 = \text{tr}(D^T F) \tag{3.7}$$

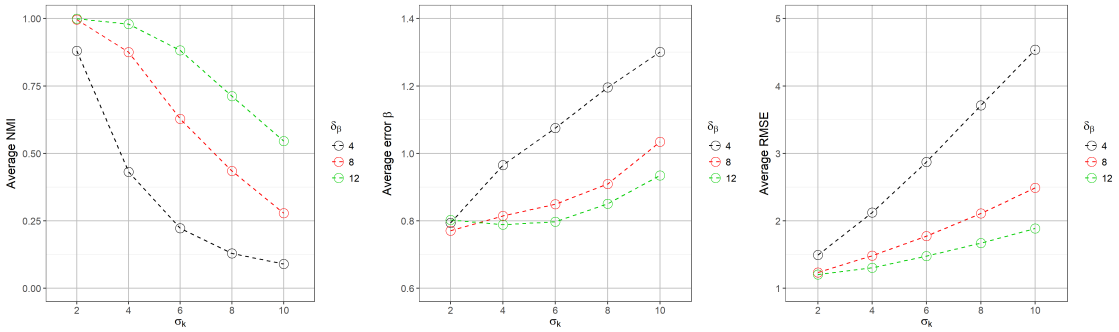
where $D = \left(\|\widehat{\beta}_k - \widehat{\beta}_\ell\|^2,\ k, \ell \in [K]\right)$ is the $K \times K$ matrix of pairwise squared distances between $\widehat{\beta}_k$s, and F is the confusion matrix between the estimated and true labels. The details for the second equality can be found in Appendix .2. Prediction RMSE is obtained by designating a hold-out (or test) set and using the trained models to predict the responses over the hold-out set. In each simulation run, 80% of the observations in each group is used for training the model, while 20% is used as hold-out set to assess the prediction power.

3.3 GMR Results

In this section, we report in detail the results from the simulation and modeling experiments. Each factor of the study is presented in a sub-section.

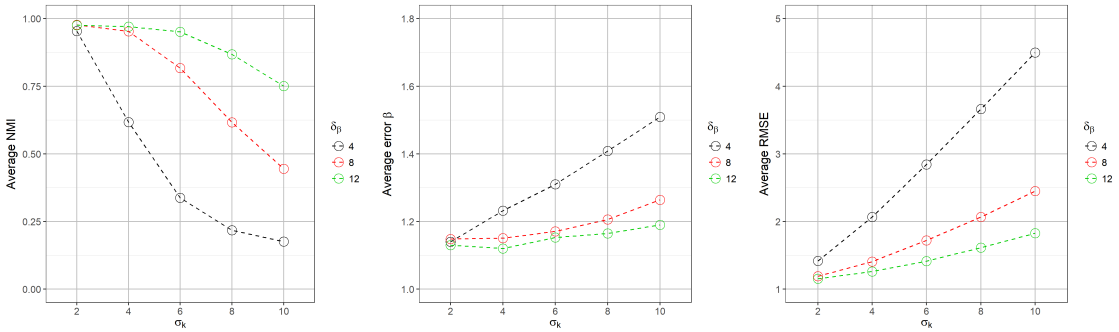
3.3.1 β-Distance (δ_β) and Noise Level (σ_k)

Figure 3.2 is the result of running the simulation and modeling experiments with $N = 100, p = 2$, and $K = 2$, a 1000 times. Referring to figure 3.2, we can see that increasing σ_k (decreasing the signal to noise ratio) causes the performance of the algorithm to decline. This is also the case with δ_β, where we can see that the more separable the true βs of the two components, the easier it gets to estimate and thus decreases the error of the algorithm. We can notice that at noise level (σ_k) of 10, and δ_β of 4, NMI (figure 3.2a) becomes close to zero, indicating that most of the times the algorithm fails to recover the true clusters. Figure 3.3 proves this conclusion by showing the same information for the case $N = 200, p = 4$, and $K = 4$.



(a) (b) (c)

Figure 3.2: The effect of δ_β and σ_k for the case $N = 100, K = 2, p = 2$; each colored line in a plot represents different vsalue of δ_β, X axis shows different values of σ_k, and y axis shows: (a) average NMI, (b) average β estimation error, (c) average RMSE for prediction, for 1000 replications of the simulation and modeling experiment



(a) (b) (c)

Figure 3.3: The effect of δ_β and σ_k for the case $N = 200, K = 4, p = 4$; each colored line in a plot represents different value of δ_β, X axis shows different values of σ_k, and y axis shows: (a) average NMI, (b) average β estimation error, (c) average RMSE for prediction, for 1000 replications of the simulation and modeling experiment

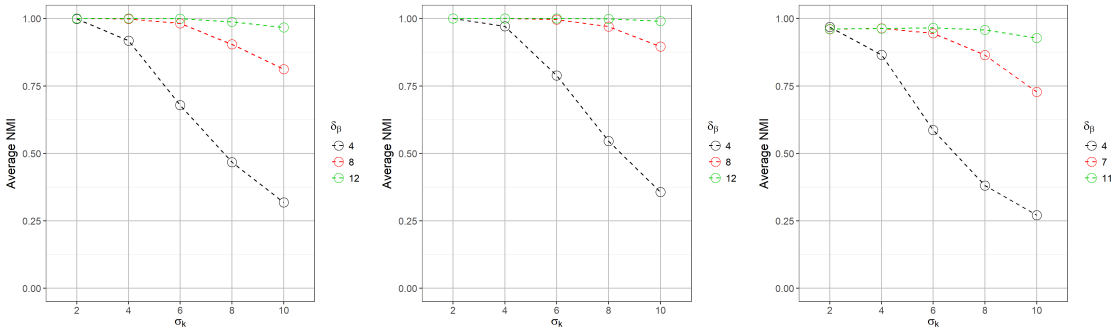
3.3.2 Dimensionality (p) and Number of Clusters (K)

Figure 3.4 shows the NMI result for different combinations of p and K, with $N = 400$. Figure 3.4a and 3.4b, compare the result when K is fixed ($K = 2$) and the dimensionality is changed from $p = 2$ in 3.4a to $p = 4$ in 3.4b. Comparing the two plots, we can see slight improvement in the case where $p = 4$. To study the effect of increasing K, we can compare figures 3.4b and 3.4c, where p is fixed ($p = 4$, and K is increased from $K = 2$ (figure 3.4b) to $K = 4$ (figure 3.4c). We can clearly see that the accuracy decreases in all cases (combinations of δ_β and σ_k). This is consistent with the fact that as the number of clusters (K) increases, it is always harder to recover true clusters.

Figure 3.5 illustrates the impact of K and p on β estimation error. By comparing the plots in figure 3.5, it is hard to find a consistent pattern for β estimation error behavior with respect to p and K. What could be noticed is that in the case where $K = 2$ and $p = 2$, the error is less sensitive to increasing the noise (σ_k). However, the error stays higher when the noise is smaller (between 2-6). In the case of $\sigma_k = 10$, the highest error belongs to the case $K = 4, p = 4$.

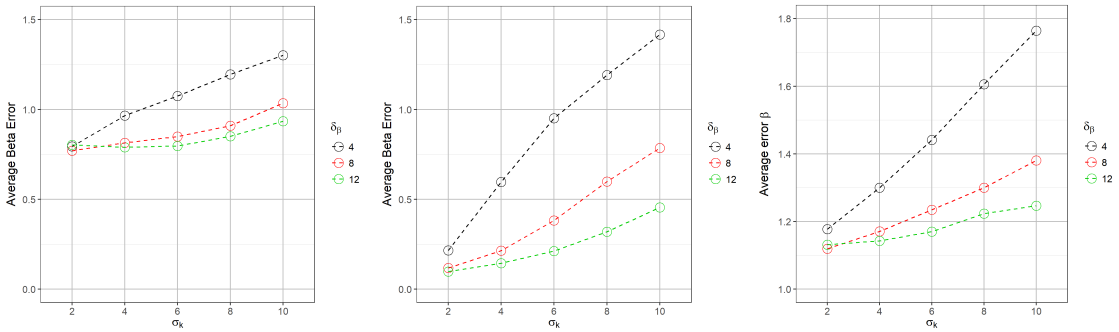
3.3.3 Number of Observations (N)

Figure 3.6 shows the average β estimation error for different number of observations (N). As mentioned earlier, with a fixed value for the number of groups (G), increasing/decreasing N causes the number of observations per group (n_r) to decrease/increase. By looking at figure 3.6, we observe that increasing N results in improving the quality of estimation (lowering β error) in all cases. Referring to figure 3.6, it is also observed that



(a) (b) (c)

Figure 3.4: The impact of K and p on NMI for the case $N = 400$; each colored line in a plot represents different vsalue of δ_β, X axis shows different values of σ_k, and y axis is average NMI for: (a) $K = 2$, $p = 2$, (b) $K2$, $p = 4$, (c) $K = 4$, $p = 4$, for 1000 replications of the simulation and modeling experiment



(a) (b) (c)

Figure 3.5: The impact of K and p on β estimation error for the case $N = 100$; each colored line in a plot represents different vsalue of δ_β, X axis shows different values of σ_k, and y axis is average NMI for: (a) $K = 2$, $p = 2$, (b) $K = 2$, $p = 4$, (c) $K = 4$, $p = 4$, for 1000 replications of the simulation and modeling experiment

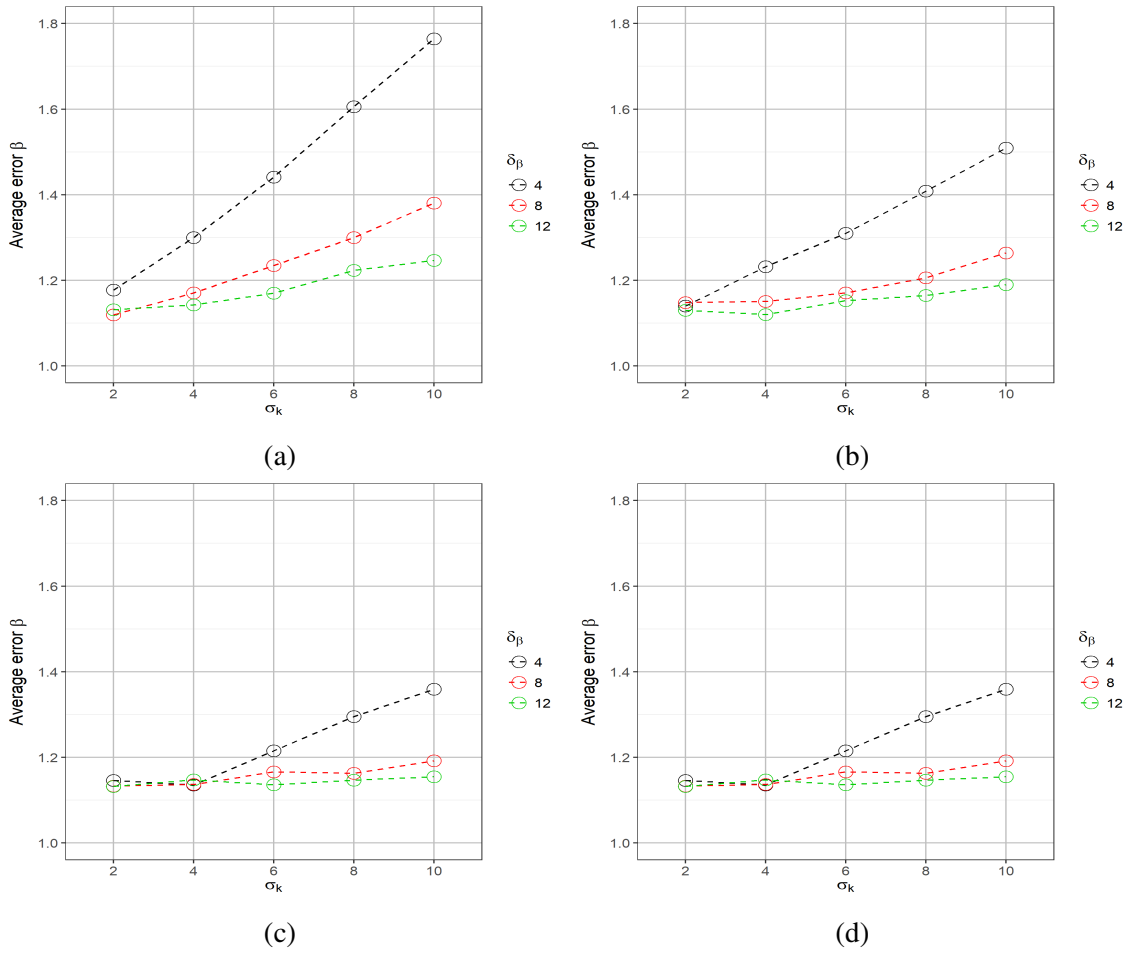


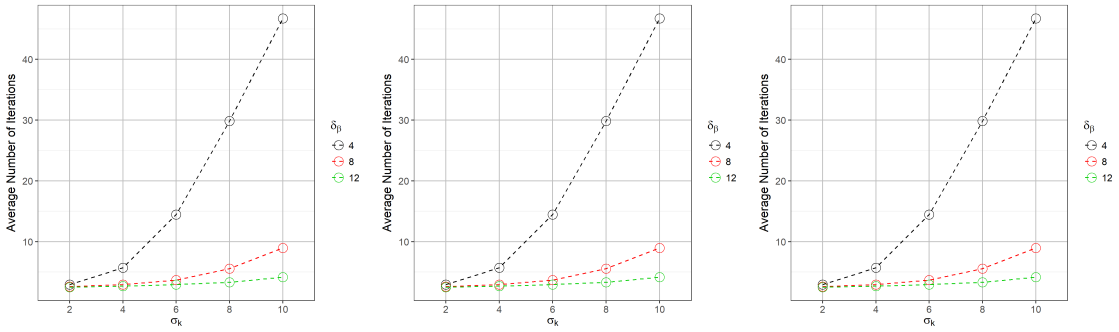
Figure 3.6: Impact of N; showing the average β estimation error for (a) $N = 100$, (b) $N = 200$, (c) $N = 400$, (d) $N = 800$, for 1000 replications of the simulation and modeling experiment

increasing N makes the results less sensitive to δ_β and σ_k.

Number of Iterations One of the important factors to determine the effectiveness of an algorithm is its speed of convergence, which determines its ability to be applied especially to high dimensional datasets. Since the speed depends on several factors such as the platform, the quality of coding, hardware, etc. it is hard to report an accurate speed for an algorithm. We report the average number of iterations for GMR convergence in each sce-

nario as an estimated indicator for the speed of the algorithm.

Stopping Criteria The stopping rule is chosen to be the relative change in posterior probability of cluster assignments ($\tau_{rk}(\hat{\theta})$ in equation (3.4)). In particular, if we call $\tau_{rk}^{(t)}$ the posterior probabilities at iteration t, then the algorithm stops when $\left\|\tau_{rk}^{(t-1)} - \tau_{rk}^{(t)}\right\|_\infty < \epsilon$, where $\|.\|_\infty$ is the infinity norm (maximum absolute row sum), or the maximum number of iterations has been reached. In our setup, ϵ is set to 10^{-6} and maximum number of iterations is set to 200. Figure 3.7 shows average number of iterations for selected scenarios.



(a) (b) (c)

Figure 3.7: Average number of iterations ($N = 800$) for: (a) $K = 2$, $p = 2$, (b) $K = 2$, $p = 4$, (c) $K = 4$, $p = 4$, for 1000 replications of the simulation and modeling experiment

Tables (3.2–3.5) provide full details of the results for the conducted simulation and modeling experiments mentioned in this section.

Table 3.2: NMI Performance

		$\delta_\beta = 4$					$\delta_\beta = 7$					$\delta_\beta = 11$				
	N \ σ_k	**2**	**4**	**6**	**8**	**10**	**2**	**4**	**6**	**8**	**10**	**2**	**4**	**6**	**8**	**10**
$K = 2; d = 2$	**100**	0.88	0.43	0.22	0.13	0.09	0.99	0.87	0.62	0.43	0.27	0.99	0.98	0.88	0.71	0.54
	200	0.98	0.71	0.40	0.25	0.15	0.99	0.98	0.88	0.68	0.54	1	0.99	0.97	0.92	0.82
	400	0.99	0.91	0.68	0.46	0.32	1	0.99	0.98	0.90	0.81	1	1	0.99	0.98	0.96
	800	1	0.99	0.98	0.90	0.75	1	1	1	0.99	0.99	1	1	1	1	1
$K = 2; d = 4$	**100**	0.93	0.49	0.21	0.13	0.09	0.99	0.93	0.73	0.48	0.32	0.99	0.99	0.93	0.8	0.64
	200	0.99	0.82	0.45	0.26	0.16	1	0.99	0.94	0.80	0.62	1	0.99	0.99	0.97	0.91
	400	1	0.97	0.79	0.54	0.35	1	1	0.99	0.97	0.89	1	1	1	0.99	0.99
	800	1	0.99	0.96	0.84	0.64	1	1	1	0.99	0.98	1	1	1	1	0.99
$K = 4; d = 4$	**100**	0.80	0.34	0.20	0.15	0.12	0.97	0.80	0.52	0.34	0.25	0.98	0.94	0.81	0.62	0.45
	200	0.95	0.61	0.33	0.21	0.17	0.97	0.95	0.81	0.61	0.44	0.97	0.97	0.95	0.86	0.75
	400	0.96	0.86	0.58	0.38	0.27	0.96	0.96	0.94	0.86	0.72	0.96	0.96	0.96	0.95	0.92
	800	0.95	0.95	0.84	0.64	0.47	0.95	0.95	0.96	0.95	0.92	0.96	0.95	0.96	0.96	0.96

Table 3.3: β Error

		β-distance = 4					β-distance = 7					β-distance = 11				
	N \ σ_k	**2**	**4**	**6**	**8**	**10**	**2**	**4**	**6**	**8**	**10**	**2**	**4**	**6**	**8**	**10**
$K = 2; d = 2$	**100**	0.79	0.96	1.07	1.19	1.42	0.77	0.81	0.85	0.90	1.03	0.8	0.78	0.79	0.85	0.93
	200	0.76	0.81	0.97	1.06	1.15	0.78	0.81	0.79	0.87	0.90	0.79	0.76	0.78	0.81	0.81
	400	0.79	0.76	0.85	0.96	1.02	0.79	0.79	0.8	0.82	0.85	0.78	0.79	0.76	0.81	0.79
	800	0.07	0.1	0.14	0.21	0.31	0.06	0.07	0.08	0.1	0.12	0.06	0.06	0.07	0.08	0.09
$K = 2; d = 4$	**100**	0.21	0.59	0.95	1.19	1.41	0.11	0.59	0.38	0.59	0.78	0.09	0.14	0.21	0.32	0.45
	200	0.14	0.32	0.63	0.88	1.06	0.09	0.15	0.21	0.32	0.47	0.08	0.11	0.14	0.18	0.25
	400	0.11	0.19	0.35	0.55	0.76	0.08	0.10	0.14	0.19	0.26	0.07	0.09	0.11	0.13	0.16
	800	0.09	1.40	0.20	0.31	0.46	0.07	0.09	0.11	0.13	0.17	0.06	0.07	0.09	0.10	0.12
$K = 4; d = 4$	**100**	1.17	1.3	1.44	1.60	1.76	1.12	1.17	1.23	1.3	1.38	1.13	1.14	1.41	1.62	1.83
	200	1.14	1.23	1.31	1.40	1.51	1.14	1.15	1.17	1.20	1.26	1.13	1.12	1.15	1.16	1.19
	400	1.14	1.13	1.21	1.29	1.36	1.13	1.13	1.16	1.16	1.19	1.13	1.14	1.13	1.14	1.15
	800	1.13	1.14	1.16	1.21	1.25	1.13	1.13	1.14	1.14	1.14	1.14	1.13	1.13	1.14	1.14

Table 3.4: RMSE Performance

		β-distance = 4					β-distance = 7					β-distance = 11				
	σ_k / N	**2**	**4**	**6**	**8**	**10**	**2**	**4**	**6**	**8**	**10**	**2**	**4**	**6**	**8**	**10**
$K=2; d=2$	**100**	1.49	2.12	2.87	3.71	4.54	1.23	1.48	1.77	2.10	2.49	1.2	1.30	1.47	1.67	1.88
	200	1.23	2.11	2.87	3.69	4.51	1.23	1.47	1.76	2.10	2.49	1.20	1.32	1.46	1.66	1.88
	400	1.46	2.11	2.88	3.68	4.52	1.25	1.48	1.78	2.13	2.48	1.23	1.32	1.46	1.66	1.89
	800	1.39	2.06	2.83	3.6	4.49	1.18	1.4	1.7	2.06	2.44	1.13	1.26	1.4	1.6	1.833
$K=2; d=4$	**100**	1.45	2.10	2.86	3.69	4.53	1.23	1.45	1.75	2.11	2.50	1.18	1.30	1.44	1.65	1.86
	200	1.45	2.10	2.87	3.68	4.51	1.23	1.45	1.75	2.10	2.48	1.19	1.30	1.45	1.64	1.86
	400	0.14	2.10	2.86	3.67	4.51	1.24	1.45	1.75	2.10	2.47	1.19	1.30	1.45	1.64	1.86
	800	1.44	2.09	2.86	3.67	4.50	1.23	1.45	1.74	2.09	2.46	1.19	1.29	1.45	1.65	1.86
$K=4; d=4$	**100**	1.41	2.07	2.85	3.67	4.50	1.18	1.41	1.72	2.07	2.45	1.14	1.25	1.41	1.62	1.83
	200	1.41	2.06	2.84	3.66	4.49	1.19	1.40	1.72	2.06	2.45	1.15	1.26	1.41	1.61	1.82
	400	1.41	2.07	2.84	3.66	4.50	1.19	1.41	1.72	2.06	2.44	1.14	1.25	1.41	1.60	1.83
	800	1.41	2.06	2.84	3.65	4.49	1.19	1.41	1.72	2.07	2.44	1.14	1.25	1.41	1.60	1.82

Table 3.5: Number of Iterations

		β-distance = 4					β-distance = 7					β-distance = 11				
	N \ σ_k	**2**	**4**	**6**	**8**	**10**	**2**	**4**	**6**	**8**	**10**	**2**	**4**	**6**	**8**	**10**
$K=2;\ d=2$	**100**	14.1	53.7	84.2	100.2	109.0	4.9	14.3	32.9	53.7	71.3	3.8	7.5	13.6	27.2	40.1
	200	6.3	27.6	60.6	82.4	101.3	3.3	6.4	14.4	29.3	44.7	3.1	3.8	6.6	11.5	19.6
	400	3.6	12.3	33.2	55.4	74.1	2.9	3.6	6.4	12.9	20.4	2.8	3.0	3.5	5.6	8.3
	800	2.7	3.4	6.8	13.9	24.7	2.5	2.8	3	3.4	4.7	2.4	2.6	2.8	2.9	3.1
$K=2;\ d=4$	**100**	11.4	42.8	65.1	72.4	73.6	4.2	12.1	25.2	43.2	55.2	3.6	5.8	12.0	19.6	30.8
	200	4.8	20.6	45.8	65.4	73.4	3.2	4.8	11.2	20.5	32.8	3.1	3.4	4.7	8.2	13.7
	400	3.2	8.8	23.3	42.5	57.5	2.9	3.2	4.7	8.6	15.7	2.8	3.0	3.2	3.8	5.7
	800	2.8	4.0	9.5	19.7	34.2	2.5	2.8	3.1	3.8	6.1	2.4	2.6	2.8	3.1	3.3
$K=4;\ d=4$	**100**	41.1	114.0	149.7	163.2	171.1	13.0	40.5	81.1	113.4	134.1	8.5	20.1	39.3	67.9	94.1
	200	18.6	74.7	121.8	149.1	162.5	7.7	18.3	42.7	72.8	103.0	9.15	10.7	18.5	33.2	52.3
	400	11.6	36.5	81.5	117.1	142.7	11.3	12.3	20.2	36.2	59.8	11.0	11.8	12.8	17.1	25.4
	800	12.7	18.1	40.5	72.8	104.6	13.0	13.2	12.0	17.8	27.7	10.1	12.6	12.7	12.5	14.0

3.3.4 Selecting Optimal Number of Components K

Selecting the number of components is a research topic that has attracted researchers for years and is still an open topic in the field of statistical machine learning. There are numerous methods introduced in the literature for determining the optimal number of clusters in a dataset (see for example [Goutte et al., 1999], [Pelleg et al., 2000], [Goutte et al., 2001], [Lletı et al., 2004], [Honarkhah and Caers, 2010]).

In the presence of independent variable(s), using Cross Validation (CV) is a simple and popular way for selecting the parameters. We setup an experiment with $\delta_\beta = (8, 12), \sigma_k = 6$, and $N = 200$, where the data was generated using a true $K^* = 4$. Then, the training set is trained using GMR to cluster the groups using $K = \{2, \ldots, 8\}$ and the hold out set (20%) of the observations from each group is predicted for each case of $K = \{2, \ldots, 8\}$ and the prediction RMSE is recorded.

The experiment is repeated 250 times and the average prediction RMSE is displayed in figure 3.3.4. Referring to figure 3.3.4, x axis holds the values of K that are used to apply the GMR ($K \in \{2, \ldots, 8\}$) to the data that is in reality generated with $K^* = 4$ components. To compare the performance with baselines, the testing data is predicted using mean (of the response y) of the training data. The result is corresponding to the case $K = 0$ in figure 3.3.4. $K = 1$ in figure 3.3.4 is the prediction RMSE for the case that a single linear regression model is fit to the training data and used to predict the response for the testing dataset. We can clearly see that the error decreases when we start applying GMR at $K = 2$ versus using the mean prediction ($K = 0$) and using a single linear model ($K = 1$), for both

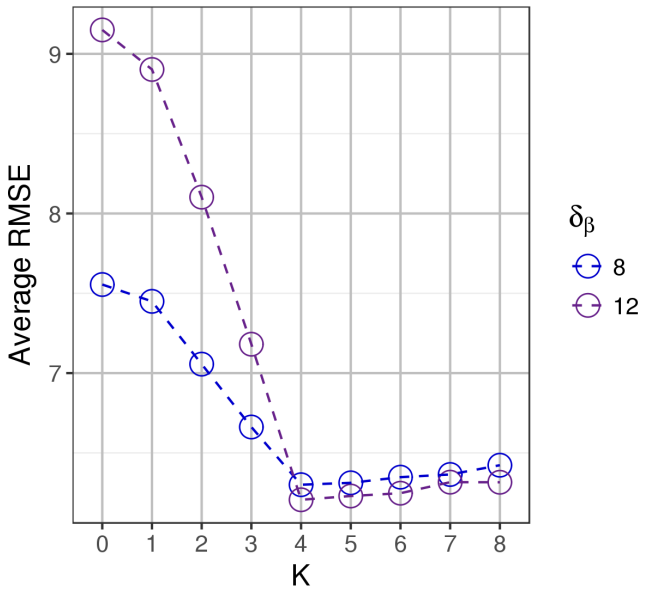


Figure 3.8: Finding the optimal value of K using cross validation: true $K^* = 4$. GMR is applied with different numbers of $K \in \{1, \ldots, 8\}$ and they are shown in x axis in the graph. $K = 0$ refers to prediction by mean while $K = 1$ is the result of prediction using a single linear regression model.

($\delta_\beta = 8$ and $\delta_\beta = 12$). The error keeps decreasing until the minimum average prediction RMSE happens when $K = 4$, where RMSE starts increasing afterwards. This plot proves the ability of the algorithm to find the optimal number of components using CV.

3.3.5 Prediction Performance

As noted earlier in chapter 3.1.3, the advantage of GMR over regular FMR is the posterior predictive density that enables us to utilize prior information about the group that a new observation is coming from. We claim that utilizing this prior knowledge can lead to a better prediction accuracy. To test the robustness of the prediction power of the model, we sample data by setting $N = 200, K = p = 4, \delta_\beta = 8$ and train the GMR using 80% of the data in each group. We then predict the remaining 20% first with usual FMR method and then using GMR prediction. Note that in usual FMR, once the model is trained

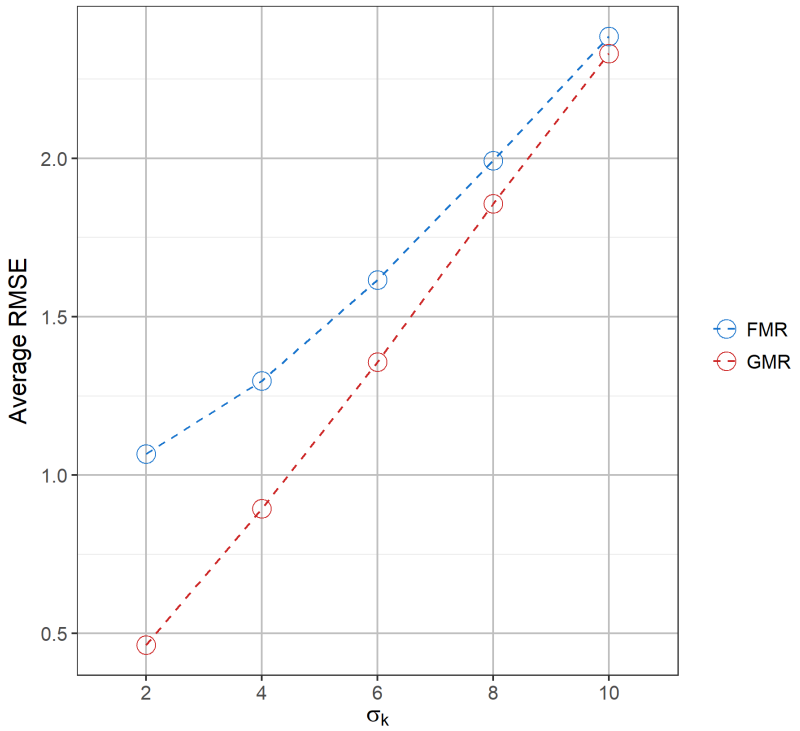
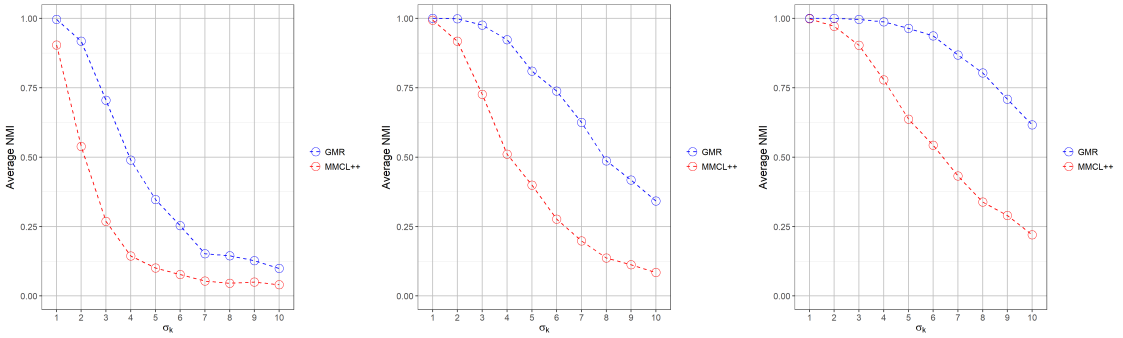


Figure 3.9: Comparing the prediction accuracy using MAP (equation 3.4) (red) versus regular FMR prediction (blue)

and the parameters of the models are estimated, new observations will be predicted using the standard mixture model rule. In our group structure setup, we obtain the parameters $\theta = (\widehat{\beta}_k, \hat{\sigma}_k, \pi_k; k \in \{1, \ldots, K\})$ in training phase. The prediction using regular FMR is $y_{new} = \sum_{k=1}^{K} \pi_k \widehat{\beta}_k^T x_{new}$. However, the prediction using predictive density obtained in 3.4 will be $y_{r,\text{new}} = \sum_{k=1}^{K} \tau_{rk}(\widehat{\theta})\, \phi_{\sigma_k}\left(y_{r,\text{new}} - \widehat{\beta}_k^T x_{,\text{new}} r\right)$. The experiment is repeated 250 times and figure 3.3.5 shows the average RMSE result (y axis is plotted for each value of σ_k). It is clear that GMR prediction using MAP outperforms the result when the prior group membership information is not used (e.g. regular FMR).

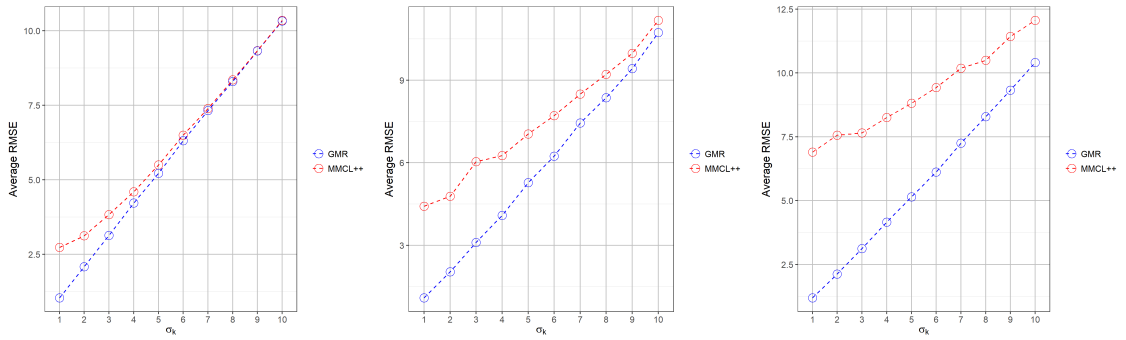


(a) (b) (c)

Figure 3.10: Comparing MMCL++ and GMR: Average NMI value for 250 simulation and modeling replications ($N = 100$) for (a) $\delta_\beta = 4$, (b) $\delta_\beta = 8$, (c) $\delta_\beta = 12$

3.3.6 Comparing GMR with *MMCL++*

To perform a comparison between our two developed algorithms: *MMCL++* and GMR, we ran a simulation and modeling experiment in a way that after generating the data according to table 3.1, both algorithms are applied to the same data and the result is recorded (we increased levels of σ_k to capture more information). The experiment is repeated 250 times for each setup. Figure 3.10 illustrates the results for the case $N = 100$. We can observe that GMR outperforms *MMCL++* in all cases in terms of correctly recovering the true labels. To compare and evaluate the prediction power of both algorithms, figure 3.11 is generated to display the average RMSE value resulted from predicting a hold out (20%) testing dataset in the same above experiment. This figure reveals that although they become very close in performance when the uncertainty in the system is very high ($\sigma_k > 5$ and $\delta_\beta = 4$), GMR performs superior to *MMCL++* for predicting new observations in all the other situations.



(a) (b) (c)

Figure 3.11: Comparing the prediction power of MMCL++ and GMR: Average RMSE value for 250 simulation and modeling replications ($N = 100$) for (a) $\delta_\beta = 4$, (b) $\delta_\beta = 8$, (c) $\delta_\beta = 12$

3.4 Conclusion

In this study, we introduced a solution to Finite Mixture of Regressions (FMR) with group structure, labeled Group Mixture Regression (GMR). We formulated the Expectation Maximization in this setup and provided the solution including the updating rules for parameter estimation. We derived a prediction density that uses prior information about the group membership of new observations to improve the prediction accuracy.

Monte Carlo simulation and modeling experiments confirm the robustness of the algorithm. Using cross validation, it successfully selected the optimal number of components, which is in general a very hard task in any clustering technique.

We ran an empirical study to compare GMR to our *MMCL++* algorithm and found that GMR outperforms *MMCL++* in both the ability to recover true clusters as well as yielding higher prediction accuracy.

CHAPTER 4: MULTI-OBJECTIVE OPTIMIZATION (MOO)

4.1 Introduction

Having a model for a specific process that depends on some parameters (predictors), it is often desirable to determine the optimal settings for each of the independent variables (parameters) so as to attain some desired process performance. Also, in case there are multiple dependent variables (y_1 and y_2), a problem arises on how to adjust the parameters in order to jointly improve the performance of both dependent variables. So, the idea is to set up an optimization problem to maximize or minimize both y_1 and y_2 (or to maximize one and minimize the other, etc.). So, this is a Multi-objective Optimization (MOO) problem under the condition of multicollinearity of the dependent variables (as well as between the dependent and independent variables).

Assume that we have a dataset with n observations and two dependent variables: y_1 and y_2, with predictors $\{x_1, x_2, ..., x_p\}, x_i \in \mathbb{R}^p$. Let X denote the matrix of covariates (the design matrix), $X \in \mathbb{R}^{n.p}$, where there are relations between $x_i's$ and (y_1, y_2). Moreover, we have to check to see if there are relations between the independent variables (x_is) themselves (i.e., strong multicollinearity), and consider (satisfy) these relations when formulating the MOO problem. This is the key requirement in formulating this problem. The reason is that without considering the multicollinearity among the independent variables, the model does not account for the fact that the independent variables may effect each other when changed, and therefore the model and in turn the process would produce incorrect re-

sults.

4.2 Deriving Recommendations under MMCL: MOO

As noted earlier, the focus of this research is to develop methods that can facilitate improvement in the performance of individual stores by relying on a data-driven approach to internal benchmarking. In particular, the goal is to identify factors driving automotive dealership performance in comparison with "similar" dealerships and relying on optimization to derive tailored recommendations. However, as noted by [Thomas et al., 1998] and others, more than one performance outcome usually needs to be considered because stores are responsible for multiple and sometimes conflicting performance measures (e.g., sales and profits). In addition, it is often the case that KPIs are competing for resources and cannot be adjusted independently at will (e.g., cash flow constraints might force the dealer to choose between adding more new vehicle sales staff or more service technicians but not both).

4.2.1 Formulating MOO

As mentioned in the previous the section, the ultimate goal is to find the optimal values for the independent parameters in order to jointly improve the dependent variables y_1 and y_2. Let $x_i \in \mathbb{R}^p$ denote an independent variable. We can setup the multi-objective optimization problem as follows:

1) *Regression: Model relationships between independent (i.e., KPIs) and dependent variables*

   Regress y_1 and y_2 on $x_1, x_2, ..., x_p$ to obtain $f_{y_1}(x_1, x_2, \ldots, x_p)$ and $f_{y_2}(x_1, x_2, ..., x_p)$.

If linear regression is employed for modeling, the result is:

$$y_1 = f_{y_1}(x_1, x_2, ..., x_p) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \epsilon_{y_1} \tag{4.1}$$

$$y_2 = f_{y_2}(x_1, x_2, \ldots, x_p) = \gamma_0 + \gamma_1 x_1 + \gamma_2 x_2 + \cdots + \gamma_p x_p + \epsilon_{y_2} \tag{4.2}$$

2) *Multicollinearity: Regress each independent variable as a function of remaining independent variables*

Again, in the case of linear regression, we will have:

$$x_i = f(x_1, x_2, \ldots, x_r, y_1, y_2) = \alpha_{i0} + \alpha_{i1} x_1 + \alpha_{i2} x_2 + \cdots + \alpha_{ip} x_p + \epsilon_{x_i} \quad \alpha_{ii} = 0 \tag{4.3}$$

3) *Optimization: Formulation*

The MOO problem can be tackled using several different approaches. The classical means of solving such problems were primarily focused on scalarizing multiple objectives into a single objective [Deb and Jain, 2014]. In many cases, there does not exist a single solution that simultaneously optimizes each objective. In that case, the objective functions are said to be conflicting, and there exists a (possibly infinite) number of Pareto optimal solutions. A solution is called nondominated, Pareto optimal, Pareto efficient or noninferior, if none of the objective functions can be improved in value without degrading some of the other objective values. There exist different solution philosophies and goals when setting and solving multi-objective optimization problems. Both exact and heuristic methods (e.g., genetic algorithms) have been

extensively investigated in the literature. See [Deb, 2001] for a good overview on the topic.

Given that our independent and dependent variables are (mostly) continuous and we have two dependent variables (y_1 and y_2), there exists a Pareto curve of optimal solutions. Let us assume that the primary interest is to maximize y_1, while satisfying an acceptable value for y_2. We can formulate the problem as follows:

$$\max \quad y_1 \tag{4.4a}$$

$$\text{s.t.} \quad y_2 \geq \tilde{y_2} \tag{4.4b}$$

$$y_2 = \gamma_0 + \gamma_1 x_1 + \gamma_2 x_2 + \cdots + \gamma_p x_p \tag{4.4c}$$

$$x_i = \alpha_{i0} + \alpha_{i1} x_1 + \alpha_{i2} x_2 + \cdots + \alpha_{ip} x_p) \quad \alpha_{ii} = 0 \quad \forall i \tag{4.4d}$$

The accuracy and effectiveness of the formulation results rely on the accuracy of the regression models. Since the regression models cannot be assumed to be perfect, constraints in equation (4.4) are relaxed to take the imperfection of the regression models into account. For this purpose, we allow slack for the regression models derived constraints proportional to $k * \sigma$, where σ denotes regression model standard error. Small values of k lead to strict constraints, i.e., strong agreement with regression models at the risk of recommendations that limit performance. To control the bounds of decision variable, we can (optionally) set bounds for the decision variables: $x_i \in \{x_i^{LowerBound}; x_i^{UpperBound}\}$. These bounds can be fine tuned to generate

the best result. Therefore, the formulation becomes:

$$\max \quad y_1 \tag{4.5a}$$

$$\text{s.t.} \quad y_2 \geq \tilde{y_2} \tag{4.5b}$$

$$\left|y_2 - (\gamma_0 + \gamma_1 x_1 + \gamma_2 x_2 + \cdots + \gamma_p x_p)\right| \leq k\hat{\sigma}_{\epsilon_{y_2}} \tag{4.5c}$$

$$\left|x_i - (\alpha_{i0} + \alpha_{i1} x_1 + \alpha_{i2} x_2 + \cdots + \alpha_{ip} x_p)\right| \leq k\hat{\sigma}_{\epsilon_{x_i}} \quad \alpha_{ii} = 0 \quad \forall i \tag{4.5d}$$

$$x_i \in \{x_i^{LowerBound}; x_i^{UpperBound}\} \quad \forall i \tag{4.5e}$$

Equation (4.5) considers the fact that the regression models obtained for y_2 and x_is are imperfect and, as mentioned earlier, the slack $k\sigma_{\cdot}$ is added to account for that. However, it assumes that the function f_{y_1} is the exact relation between the y_1 and $x_i, i \in \{1, \ldots, p\}$. In our formulation, we have assumed that $f_{y_1}(x_1, x_2, ..., x_p)$ is also a result of linear regression. Thus, we also have to account for imperfection of f_{y_1} linear model. To achieve this goal, we introduce an intermediate variable t and

define $t = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$ to arrive at the final formulation:

$$\max \quad t \tag{4.6a}$$

$$\text{s.t.} \quad y_2 \geq \tilde{y}_2 \tag{4.6b}$$

$$\left| t - (\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p) \right| \leq k\hat{\sigma}_{\epsilon_{y_1}} \tag{4.6c}$$

$$\left| y_2 - (\gamma_0 + \gamma_1 x_1 + \gamma_2 x_2 + \cdots + \gamma_p x_p) \right| \leq k\hat{\sigma}_{\epsilon_{y_2}} \tag{4.6d}$$

$$\left| x_i - (\alpha_{i0} + \alpha_{i1} x_1 + \alpha_{i2} x_2 + \cdots + \alpha_{ip} x_p) \right| \leq k\hat{\sigma}_{\epsilon_{x_i}} \quad \alpha_{ii} = 0 \quad \forall i \tag{4.6e}$$

$$x_i \in \{x_i^{LowerBound}; x_i^{UpperBound}\} \quad \forall i \tag{4.6f}$$

In equation (4.6), constraints (4.6c–4.6e) are designed to take the imperfection of the regression models into account and constraints (4.6f) are optional and limit the decision variables to practical bounds.

4.3 Synthetic Experiments

In this section, a simulation study is performed to evaluate the proposed MOO formulation. Synthetic data is generated to create different scenarios, enabling us to study the behavior of the algorithm under different scenarios.

4.3.1 Experiment Setup

Denote $\epsilon_{\cdot}$ as a random noise generated by drawing a value from a $\mathcal{N}(o, 1)$. Using the freeware *R* (see[R Core Team, 2016]), 1000 observations are generated according to the following setup:

(1) x_1, x_2, x_3, x_4 are first drawn from $\mathcal{N}(o, 1)$

(2) $x_2 = 0.5x_1 + 0.3x_4 + \epsilon_{x_2}$

(3) $x_3 = -0.5x_1 + 0.01x_2 + \epsilon_{x_3}$

(4) $x_4 = -0.02x_1 + 0.4x_2 + 0.3x_3 + \epsilon_{x_4}$

(5) $y_1 = 0.5x_1 + 0.1x_2 - 0.5x_3 - 0.18x_4 + \epsilon_{y_1}$

(6) Calculate y_2 to generate the following scenarios:

(a) $y_2 = -0.9x_1 + 1.2x_3 + 0.3x_4 + \epsilon_{y_2}$ to get a correlation of -0.55 between y_1 and y_2

(b) $y_2 = -0.6x_1 - 0.08x_2 + 0.3x_4 + \epsilon_{y_2}$ to get a correlation of -0.39 between y_1 and y_2

(c) $y_2 = -0.2x_1 - 0.1x_2 + 0.2x_3 + 0.3x_4 + \epsilon_{y_2}$ to get a correlation of -0.25 between y_1 and y_2

(d) $y_2 = -0.35x_1 - 0.05x_4 + \epsilon_{y_2}$ to get a correlation of -0.1 between y_1 and y_2

Figure 4.1 shows the correlation between all the variables for each scenario. Beside the correlation between the dependent and independent variables, you can also observe that there is strong multicollinearity between the dependent variables (e.g. 0.48 correlation between x_1 and x_2 in figure 4.3b.

4.3.2 Results

The simulation is run for different values of $k \in \{0.75, 1, 1.25, 1.5\}$ (in constraints 4.6c–4.6e). By changing the lower bound for $\tilde{y_2}$ (constraint (4.6b)), the pareto optimal solution is generated for each scenario and each value of k. Figure 4.2 shows the pareto optimal solution for different scenarios.

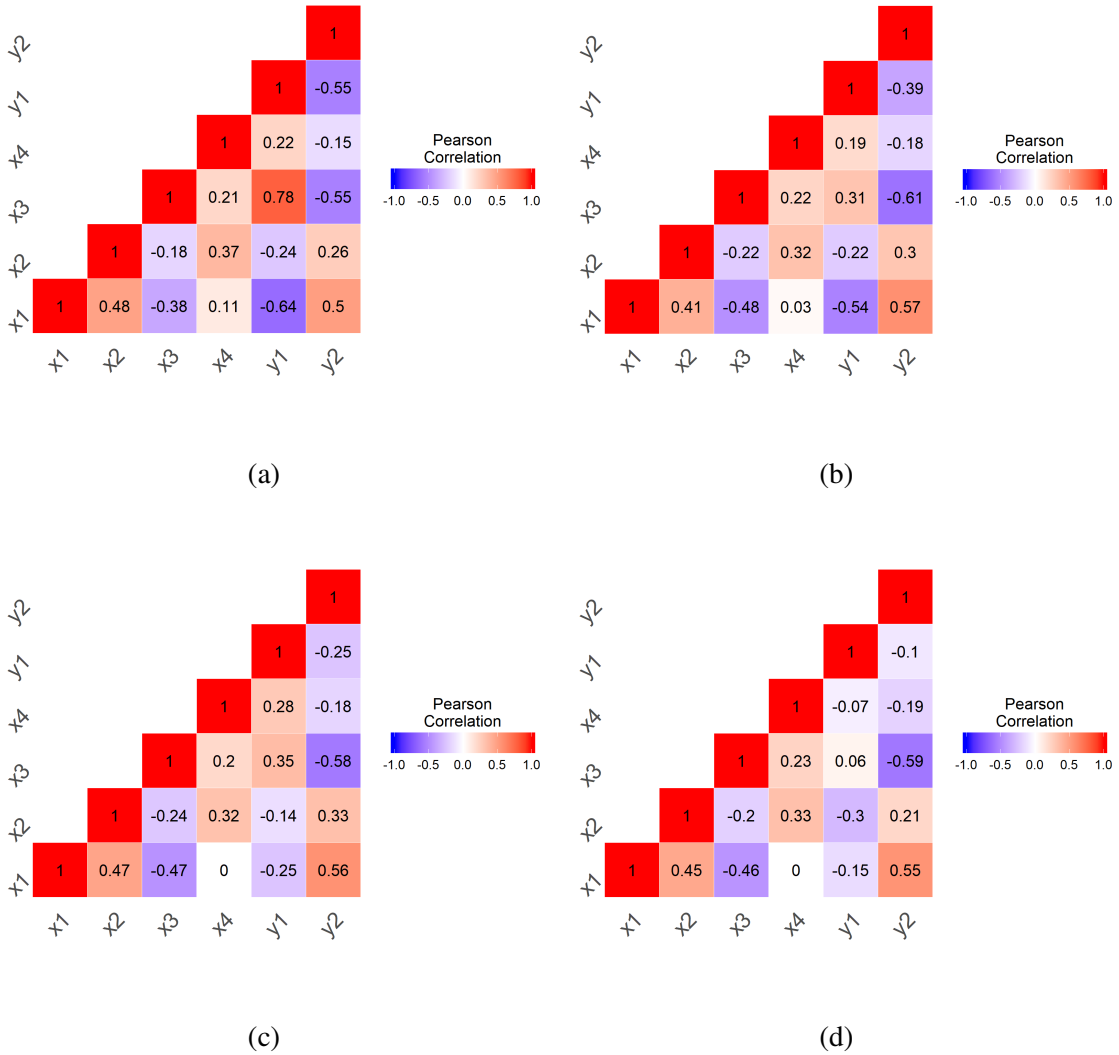


Figure 4.1: Correlation plot for variables in (a) scenario a, (b) scenario b, (c) scenario c, and (d) scenario d

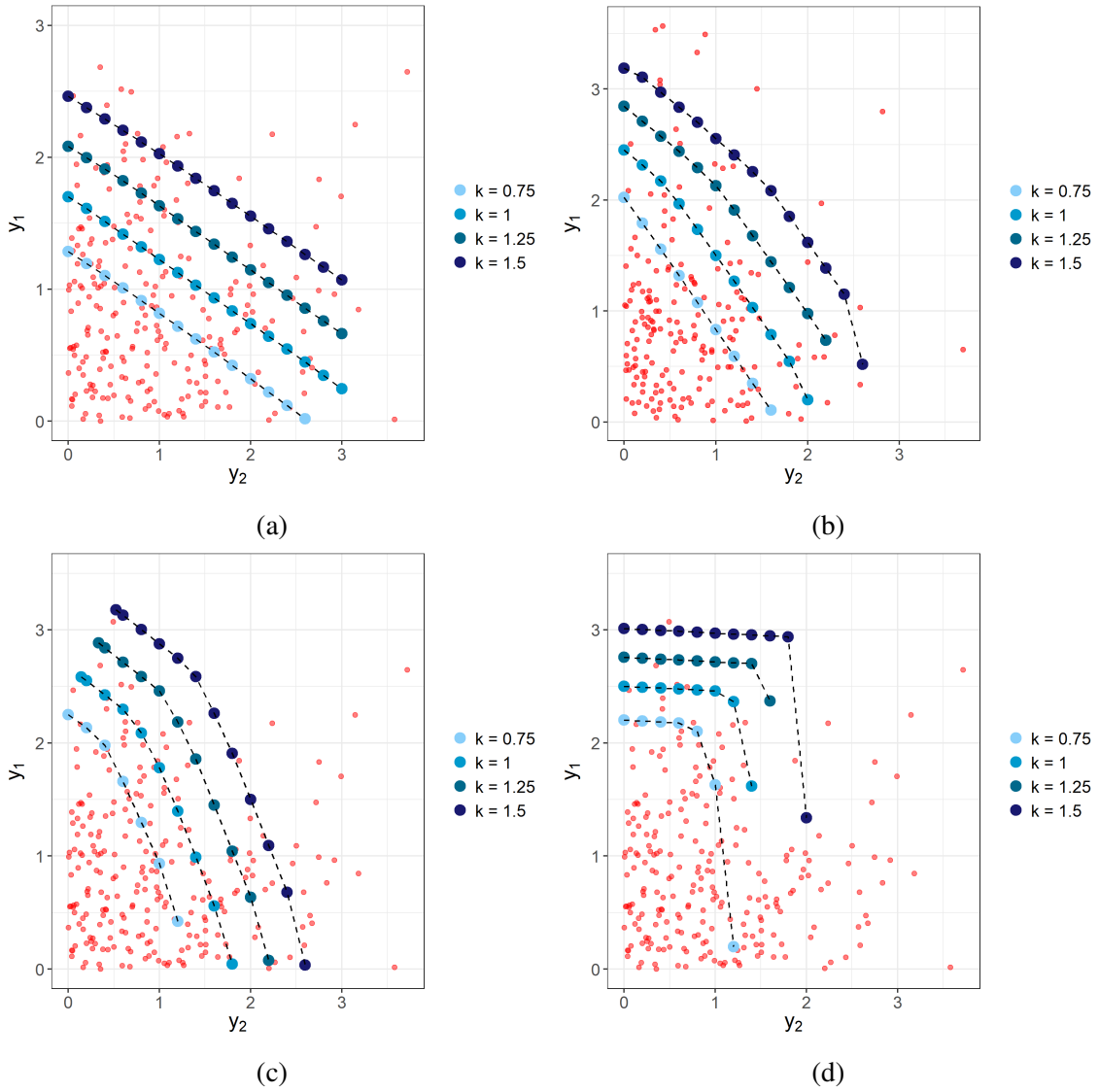


Figure 4.2: Pareto optimal solution for: (a) scenario a, (b) scenario b, (c) scenario c, and (d) scenario d

Referring to figure 4.2, we observe two extreme situations for scenarios a and d(high and low negative correlation). In scenario a, the pareto is almost linear with slope -0.5, indicating equal trade off between $y_1$andy_2. In scenario d, we can see that there is a very slow decrease in the optimal value of y_2 when increasing y_1 (the main objective). This means that when the two independent variables are not correlated (or weakly correlated), there is a great opportunity to improve one of the objective functions (e.g. y_2) without sacrificing the other objective. In scenarios b and c, the rate of change for y_1 is faster when $\tilde{y_2}$ is changed. We can also observe how the value of k can shift the pareto solution. With high values of k, we allow more slack into constraints 4.6c–4.6e. This causes the pareto solution to move up.

Now that we have obtained an optimal pareto solution, we can solve the optimization problem with fixed value for k and $\tilde{y_2}$ to obtain the optimal values for each of the independent variables. Figure 4.3 shows a solution for scenarios $a - d$, with $k = 1.1$, and $\tilde{y_2} \in \{0.5, 1, 1.5\}$.

4.4 Conclusion

In this study, we propose a Multi-objective optimization (MOO) method for obtaining the optimized values in the presence of multicollinearity. The formulation also assumes that there is negative correlation (trade off) between two objective functions. The situation may arise in different real world problems such as the trade off between spending money on advertising by a firm and the returned profit resulting from advertisements. Our formulation is unique in the sense that the relation between the decision variables as well as the objective function are statistical and are assumed to be estimated (they are imperfect). By

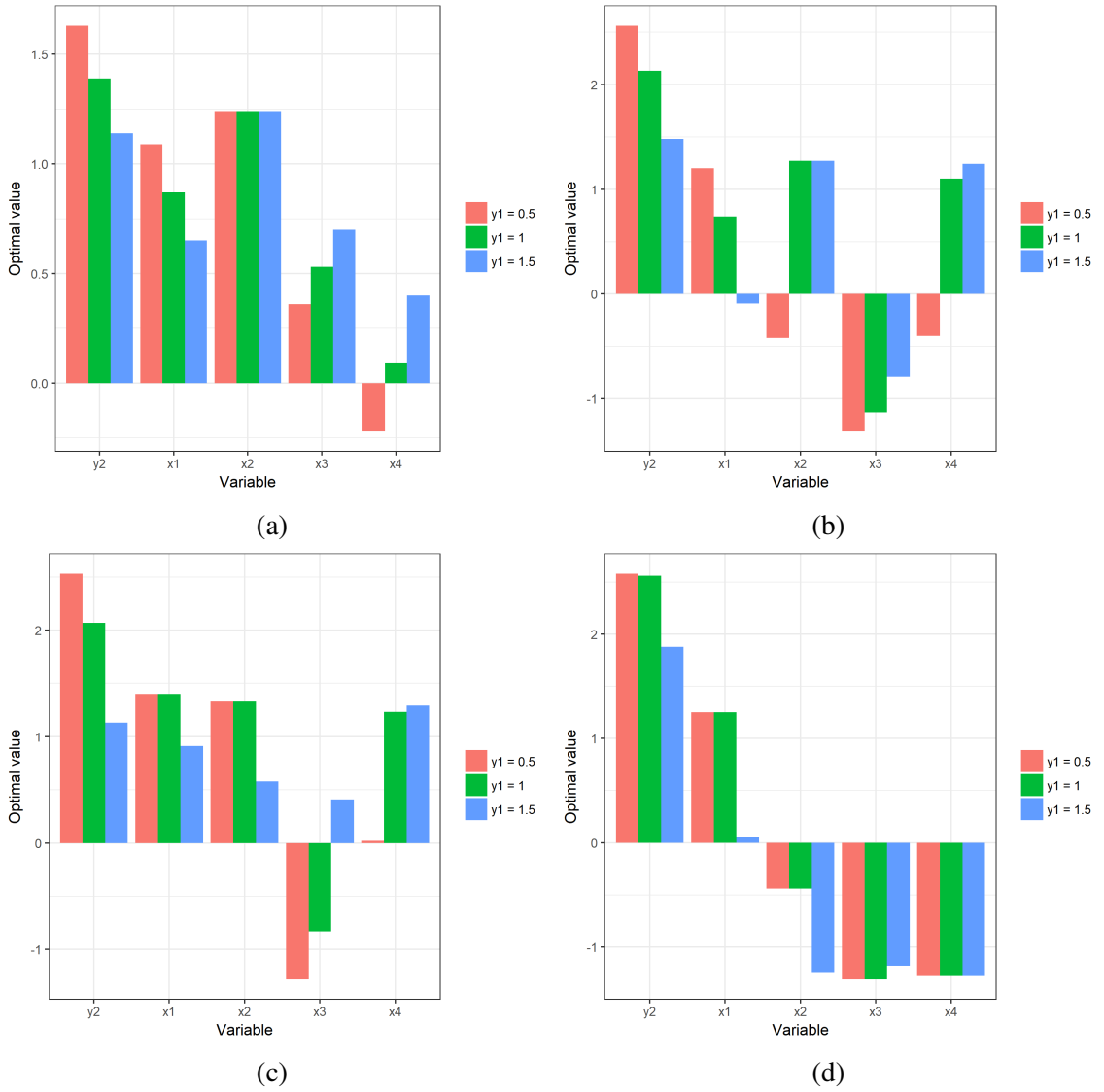


Figure 4.3: Optimal values for variables resulting from solving MOO: (a) scenario a, (b) scenario b, (c) scenario c, and (d) scenario d

solving MOO, one can obtain optimal values for decision variables that control both objective functions, while honoring the relations between decision variables themselves. The proposed method can be extended relatively easily for the case of more than two objectives.

By running a synthetic data experiment, we demonstrated different scenarios that could arise in real world problems and showed how the solution changes with regards to changing the parameters of the formulation.

CHAPTER 5: DERIVING RECOMMENDATIONS FOR DEALERSHIPS

5.1 Introduction

In this chapter, we present the results from applying the proposed methodologies (MMCL, GMR, MOO) to a real-world problem in the retail industry. We show how we can utilize these methodologies in order to provide guidelines and recommendations to improve the performance of retail stores, in particular, automotive dealerships. We first apply the two proposed algorithms to fit finite mixture of regression (FMR) models to the dealership dataset in order to segment/cluster stores while accounting for similarity in store performance dynamics. The objective is to cluster the stores into a number of smaller homogeneous store groups for benchmarking and deriving more effective recommendations. For example, it might be inappropriate to benchmark a rural dealer with an urban dealer in a large metropolitan city. This is above and beyond the normal practice of examining stores based on regional location. For example, it is a common practice in the U.S. for the automotive OEMs to look at the continental U.S. as several major regions (e.g., North-East, Midwest, South-East, South-West, and West) due to significant differences in weather and other purchasing patterns. While we too recommend regional analysis, there is still room for improving the performance modeling by further clustering the regional stores into a number of smaller homogeneous store groups.

We also demonstrate how to use the results of clustering and utilize them in the pro-

posed MOO (Chapter 4) to derive tailored recommendations and provide guidelines for the management on how to adjust the KPI levels in order to improve the store performance.

5.2 Dealership Dataset

For reasons of confidentiality, we are not able to reveal the full details about the dataset. The dataset made available consists of several thousand (3,074) dealerships, with consecutive monthly financial data (observations) for each dealer spanning five years (60 observations per dealer). Figure 5.1 shows the dealer network across the United States. It shows how the dealers are distributed and grouped into five regions: Northeast (yellow), Southeast (red), Great Lakes (blue), Central (gray), and West (black). There were 281 KPIs (independent variables) in the monthly financial documents deemed important by the domain experts. There were a number of missing entries in the financial documents and the resulting dataset. The missing values were imputed using *matrix completion via soft thresholding SVD* technique, using the "*softImpute*" package in R [Hastie and Mazumder, 2015]. The variables are standardized to carry a mean of zero and standard deviation of one (to be comparable).

To prepare the data for application of MMCL and GMR, the observations for all the dealers are first aggregated to construct the design matrix $X \in \mathbb{R}^{3074\times 281}$. Since the data for each dealership is generated for each month, we checked for trend and seasonality for each dealer and found no evidence that there exists trend or seasonality between the consecutive months. The reason is that the KPIs are constructed in a way that the trend and seasonality are absorbed by a special way of normalization. The observations for each dealer are given a group ID to represent the groups for the purpose of MMCL and GMR.

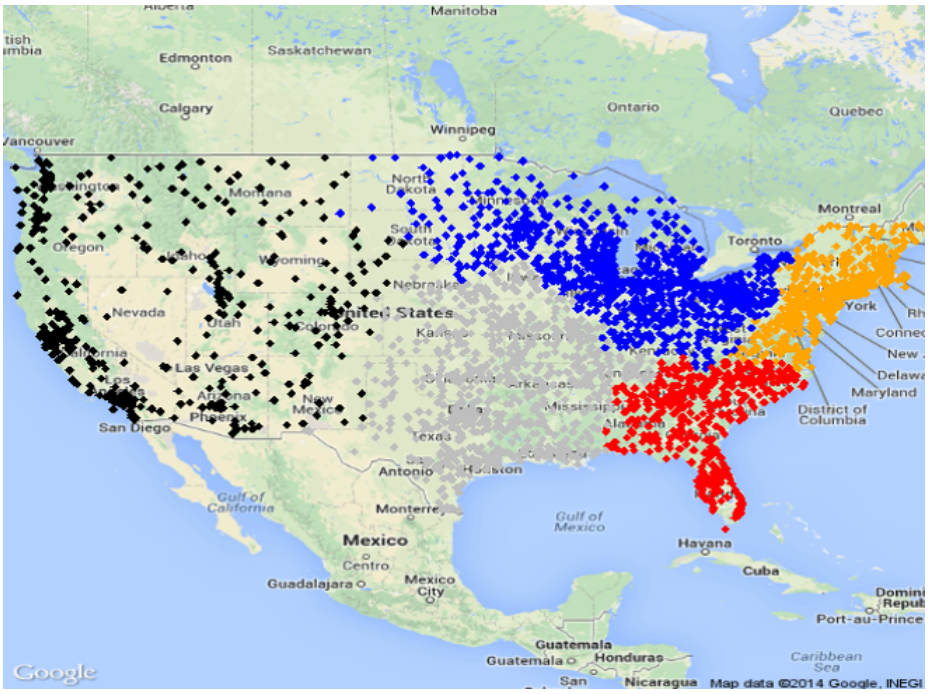


Figure 5.1: Network of OEM dealerships in the U.S. Grouped into five regions: Northeast (golden), Southeast (red), Great Lakes (blue), Central (gray), and West (black)

5.2.1 Applying MMCL and GMR to Dealership Performance Problem

We focus here on applying the proposed *MMCL* and GMR methods for modeling the productivity of automotive dealerships across the U.S. for a particular OEM. Because of the large size of the dataset and specially the large number of predictors, Least Absolute Shrinkage and Selection Operator (LASSO) [Tibshirani, 1996] is used for regression modeling of both the sales as well as the profitability of each dealership, for each month. As discussed in Section 2.3, the main parameters should be selected before running the algorithm. In the case of *MMCL* (and *MMCL++*), the parameters are: number of clusters K, selection of K dealers for initializing the clusters, and LASSO regularizer (λ). To find

the best values for these parameters, the data is split into training (first four years for each dealer), and testing (financial data from last year). The parameters are then selected using cross-validation (CV), by evaluating the quality of the models on predicting the testing data. The parameters that produced the best result (highest R^2 value) on the testing data are selected. Since *MMCL* is a heuristic method and the results highly depend on a good starting points (initial dealers), it is a advised to run the algorithm several times for each case (e.g. fixed λ and K) and pick the initial groups (dealers) that produce the best result. Once the parameters are selected, the *MMCL* algorithm is applied to the dataset. It is observed that in most cases, the algorithm converges in less than 15 iterations.

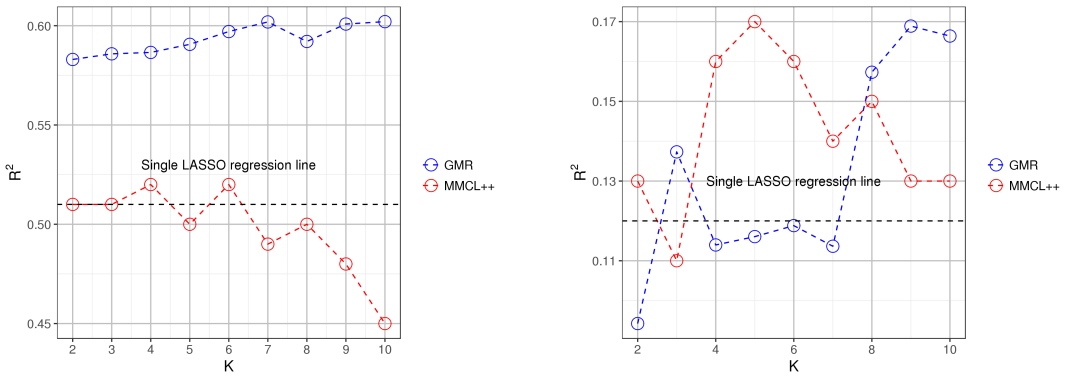
As for GMR, the only parameter that should be optimized before applying the algorithm is the number of clusters K. As demonstrated through sythetic simulation experiments in section 3.3.4, GMR is proved to properly select the true K. The same process is applied to the dealership dataset to find the best K.

The result is presented in table 5.1. Referring to table 5.1, the reported R^2 is the value obtained by predicting the testing dataset using each model/algorithm. It is reported for both Profitability (the concern of dealership) and Sales Effectiveness (OEM's main objective). The parameter λ (LASSO regulirizer) is the one that produced the highest R^2 value on the testing dataset. We are displaying the result of using three proposed algorithms: *MMCL*, *MMCL++*, and GMR for $K \in \{2, \ldots, 10\}$. The case where $K = 1$ refers to fitting a single linear regression to the training dataset. Again, because of the large size of the data, a LASSO regression is used in this case. The parameter λ is also optimized using CV.

As the results suggest,GMR has improved the accuracy for predicting both profitability and sales effectiveness. It was able to achieve a R^2 value of 0.6 using $K = 9$ and $K = 10$, whereas a single model's R^2 is 0.51, a 9% improvement. The highest R^2 that *MMCL* was able to achieve for profitability is 0.51, equal to what is achieved by a single model. *MMCL++* was able to slightly improve the result to obtain R^2 value of 0.52 (with $K = 4$ and $K = 6$). In the case of SE, a single model is able to produce a R^2 value of 0.12. However, GMR was able to improve this value to 0.17 (41% improvement) with $K = 9$. *MMCL* and *MMCL++* also produced the same result with $K = 5$. This result also suggests that there is heterogeneity among the dealers and by clustering them, one can improve the analysis and generate better recommendations to dealers for improving their performance.

Figure 5.2 summarizes the result on one plot. Since *MMCL++* has been equal or superior in performance compared to *MMCL*, we are only showing *MMCL++* versus GMR. Reviewing figure 5.2, we can conclude that if GMR is used, we should ideally cluster the dealers into 9 groups where the models show highest R^2 value for both Profitability and Sales Effectiveness. In case of modeling with *MMCL++*, it is best to partition the dealers into 4 clusters.

It should be mentioned that in large datasets such our dealership problem, *MMCL* approach is computationally more expensive. This is in general true when the number of groups is large, because the algorithm has to extract, model, and evaluate the results of all groups in every iteration. This becomes more problematic if we know that we have to find the parameters of the models as well as best starting point (e.g. initial groups) using CV, because the algorithm has to be applied several times to find the best result. This is not



(a) (b)

Figure 5.2: Results from applying *MMCL++*, and GMR on dealership dataset with two dependent variables: (a) Profitability and (b) Sales Effectiveness. The horizontal black dashed line is R^2 value for a single LASSO model.

the case for GMR, for it had no problem handling and producing the result of such a large dataset in seconds.

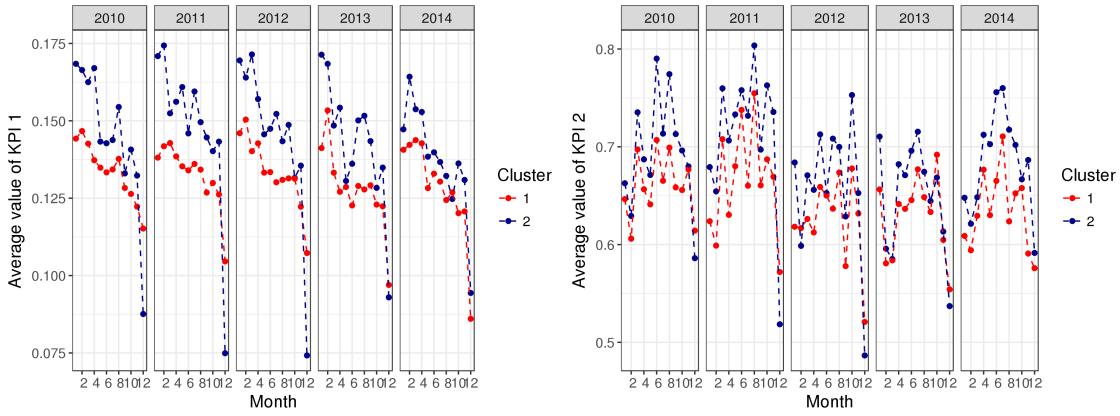
5.2.2 Assessing the Clusters

To evaluate the clusters resulted as the output of GMR, we applied GMR to a subset of the data (a sub-group defined by domain experts). The number of clusters K is set to 2. The following plots are produced to visually evaluate the effectiveness of the formed clusters. Figure 5.3 shows the average value for two of the most important KPIs (with highest regression model coefficients), plotted for two clusters (cluster 1 red, cluster 2 blue). Note that independent variables are standardized to carry a mean of zero and standard deviation of one (to be comparable).

As shown in figure 5.3, the average value of KPI #1 in figure 5.3a is clearly different between the two clusters and cluster 2 (red) tends to contain the dealers that have a smaller value in that particular KPI. In the case of the KPI #2 in figure 5.3b, there are some months

Table 5.1: Result of applying *MMCL*, *MMCL++*, and GMR to dealership data

		Profitability		**Sales Effectiveness**	
# of Clusters (K)	Algorithm	Model Parameters	R^2	Model Parameters	R^2
1	Single LASSO	$\lambda = 0.036$	0.51	$\lambda = 0.051$	0.12
2	MMCL	$\lambda = 0.005$	0.51	$\lambda = 0.005$	0.10
	MMCL++	$\lambda = 0.001$	0.51	$\lambda = 0.001$	0.13
	GMR	NA	0.58	NA	0.09
3	MMCL	$\lambda = 0.015$	0.5	$\lambda = 0.011$	0.10
	MMCL++	$\lambda = 0.011$	0.51	$\lambda = 0.011$	0.11
	GMR	NA	0.58	NA	0.13
4	MMCL	$\lambda = 0.031$	0.51	$\lambda = 0.015$	0.13
	MMCL++	$\lambda = 0.031$	0.52	$\lambda = 0.011$	0.16
	GMR	NA	0.58	NA	0.11
5	MMCL	$\lambda = 0.005$	0.5	$\lambda = 0.011$	0.17
	MMCL++	$\lambda = 0.005$	0.5	$\lambda = 0.015$	0.17
	GMR	NA	0.59	NA	0.11
6	MMCL	$\lambda = 0.005$	0.5	$\lambda = 0.015$	0.15
	MMCL++	$\lambda = 0.011$	0.52	$\lambda = 0.021$	0.16
	GMR	NA	0.59	NA	0.11
7	MMCL	$\lambda = 0.011$	0.49	$\lambda = 0.011$	0.14
	MMCL++	$\lambda = 0.021$	0.49	$\lambda = 0.001$	0.14
	GMR	NA	0.6	NA	0.11
8	MMCL	$\lambda = 0.021$	0.48	$\lambda = 0.025$	0.14
	MMCL++	$\lambda = 0.031$	0.5	$\lambda = 0.031$	0.15
	GMR	NA	0.59	NA	0.16
9	MMCL	$\lambda = 0.035$	0.44	$\lambda = 0.025$	0.13
	MMCL++	$\lambda = 0.011$	0.48	$\lambda = 0.031$	0.13
	GMR	NA	0.6	NA	0.17
10	MMCL	$\lambda = 0.025$	0.45	$\lambda = 0.011$	0.13
	MMCL++	$\lambda = 0.021$	0.45	$\lambda = 0.005$	0.13
	GMR	NA	0.6	NA	0.16



(a) (b)

Figure 5.3: Assessing the clusters formed by GMR in the KPI space. The average value for each month and year is displayed for Cluster 1 (red) and Cluster 2 (blue) (a) KPI #1 (b) KPI #2

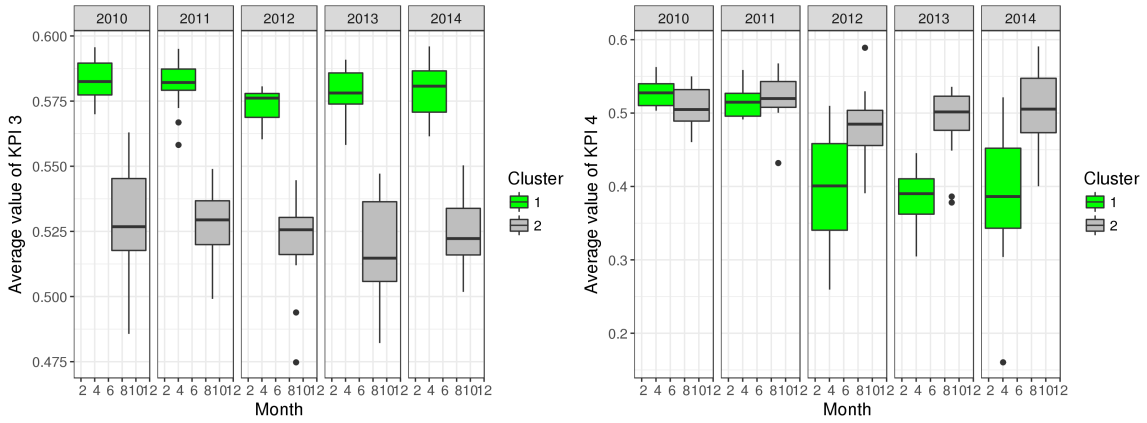
that the two clusters have overlapped, but the two clusters still seem to be different. Figure 5.4 shows box plots for two other important KPIs, separated by the clusters. This figure also proves the effectiveness of GMR in forming two clusters with members with different KPI ranges.

5.2.3 Applying MOO for Dealership Performance Improvement

As noted earlier, there are two dealer performance characteristics of interest: Profitability (P: y_1) and Sales Effectiveness (SE: y_2). It is desired that both y_1 and y_2 be maximized for every dealer to improve the profitability of the dealership and satisfy the needs of the OEM in selling more new vehicles. We call our design matrix $X \in^{n \times p}$

To apply the MOO, we first regress P (y_1) against X to obtain:

$$P = f_P(x_1, \ldots, x_p) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \epsilon_P \tag{5.1}$$



(a) (b)

Figure 5.4: Box Plot for assessing the clusters formed by GMR in the KPI space. Values for each month and year is displayed for Cluster 1 (green) and Cluster 2 (gray) (a) KPI #3 (b) KPI #4

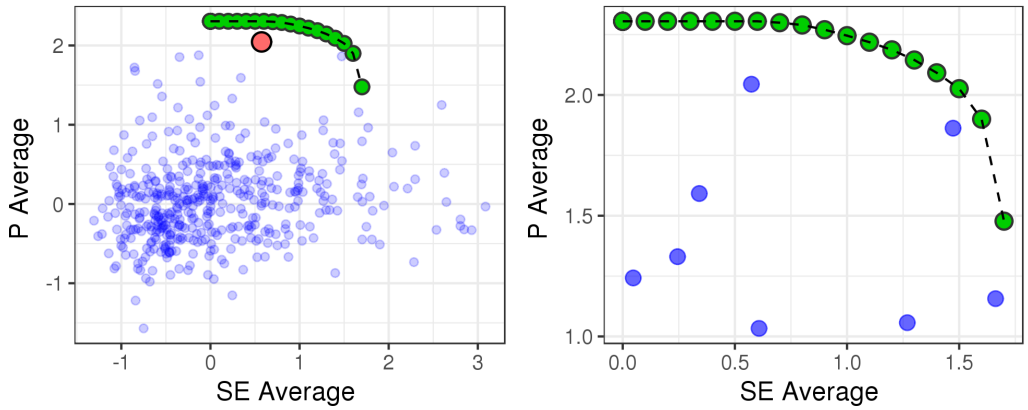
where $f_P(x_1, \ldots, x_p)$ is the objective function that we want to maximize (ignoring the error term ϵ_P). We then regress y_2 against X to obtain the constraint that explains the relation between y_2 and x_is:

$$SE = f_{SE}(x_1, x_2, \ldots, x_p) = \gamma_0 + \gamma_1 x_1 + \gamma_2 x_2 + \cdots + \gamma_p x_p + \epsilon_{SE} \tag{5.2}$$

Also, to guarantee that SE stays in the accepted range enforced by OEM, we add the following constraint:

$$SE \geq \tilde{SE}$$

Lastly, we regress each x_i against other x_js ($i \neq j$), to account for the multicollinearity constraints. As explained in Section 4.2, to consider the fact that the regression models are imperfect, we allow a slack of $k * \sigma_\epsilon$ for each regression model. The client also provided



(a) (b)

Figure 5.5: Pareto optimal frontier for a specific dealership group: (a) Blue dots report average SE and P for dealers; benchmark dealer is shown in red. (b) Zoomed-in region of Pareto optimal frontier

valid bounds for each of the predictor variables but are not reported here for confidentiality.

Assuming $t = f_P(x_1, \ldots, x_p)$, the final formulation takes the following form:

$$\max \quad t \tag{5.3a}$$

$$\text{s.t.} \quad SE \geq \tilde{SE} \tag{5.3b}$$

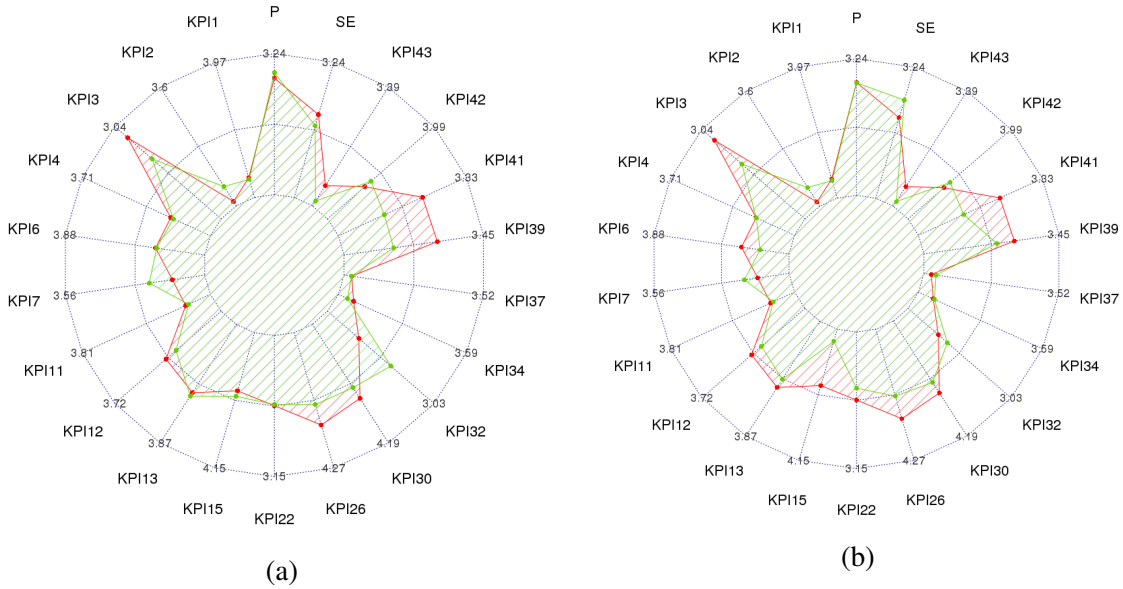
$$\left|P - (\beta 0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p)\right| \leq k\hat{\sigma}_{\epsilon_P} \tag{5.3c}$$

$$\left|SE - (\gamma_0 + \gamma_1 x_1 + \gamma_2 x_2 + \cdots + \gamma_p x_p)\right| \leq k\hat{\sigma}_{\epsilon_{SE}} \tag{5.3d}$$

$$\left|x_i - (\alpha_{i0} + \alpha_{i1} x_1 + \alpha_{i2} x_2 + \cdots + \alpha_{ip} x_p)\right| \leq k\hat{\sigma}_{\epsilon_{x_i}} \quad \alpha_{ii} = 0 \quad \forall i \tag{5.3e}$$

5.2.4 Generating Pareto Optimal Frontier

To construct the Pareto optimal front for each dealership group, the formulation above is solved repeatedly by changing the value of $\tilde{SE}$ to obtain the Pareto optimal points for P and SE, as reported in Figure 5.5a. This result identifies what is potentially possible in terms



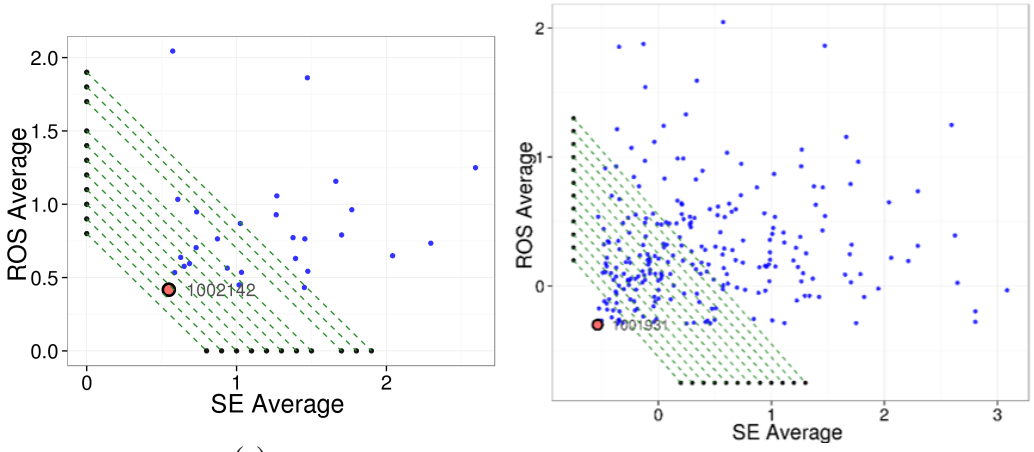
(a) (b)

Figure 5.6: Comparing derived recommendations for a reference dealership group (green) with the operations of a well performing dealer (red). (a) $\tilde{SE} = 1.0$ (b) $\tilde{SE} = 1.5$

of performance for the dealers within the reference group. As expected, the frontier also reveals the trade-off between how much profit a dealer can generate (using all the potential resources such as new and used vehicle sales, service, body shop, parts, etc.) versus new vehicle sales (not as profitable these days with respect to other dealership operations such as service and used vehicle sales).

5.2.5 Assessing the Quality of Recommendations Derived through MOO

To further assess the effectives of the proposed methods, we compared the operational signature of a high performance dealership with the recommendations derived through MOO, as reported in Figure 5.5a. Radar charts are generated for some of the important KPIs (x_is) for different values of $\tilde{SE}$ (see Figure 5.6). The reasonable agreement between the two sets of KPI values between the high performance dealer and the derived recommendations further validate the effectiveness of our proposed algorithm. For example, it can



(a) (b)

Figure 5.7: Gradual improvement path for (a) a successful dealer, and (b) a weak dealer

be seen that if a dealer in the reference group wants to achieve a better SE (on average), they have to on average lower KPI8 and KPI16, and increase KPI6. The recommended KPI levels can be compared with each dealer within the group in order to show the potential strengths and deficiencies of that dealer. It provides tailored guidance for the dealership management on how to manage and operate their business in order to simultaneously please the OEM by selling new vehicles (keep SE in an acceptable range) and also increase their profits. Figure 5.8 shows how the optimal KPI values change with changing $\tilde{SE}$.

5.2.6 Gradual Improvement Paths for Dealers

The formulation shown in equation 5.3 while satistying the constraints, maximizes the objective function (profitability) as much as possible. This means that we provide the same set of recommendations to a weak dealer (i.e. a dealer on bottom left corner in figure 5.5a and a strong and successful dealer ((i.e. a dealer on top right corner in figure 5.5a to maximize their profitability to the same amount. In reality however, it may not be realistic for weak dealers to adjust their KPIs according to optimal values and achieve

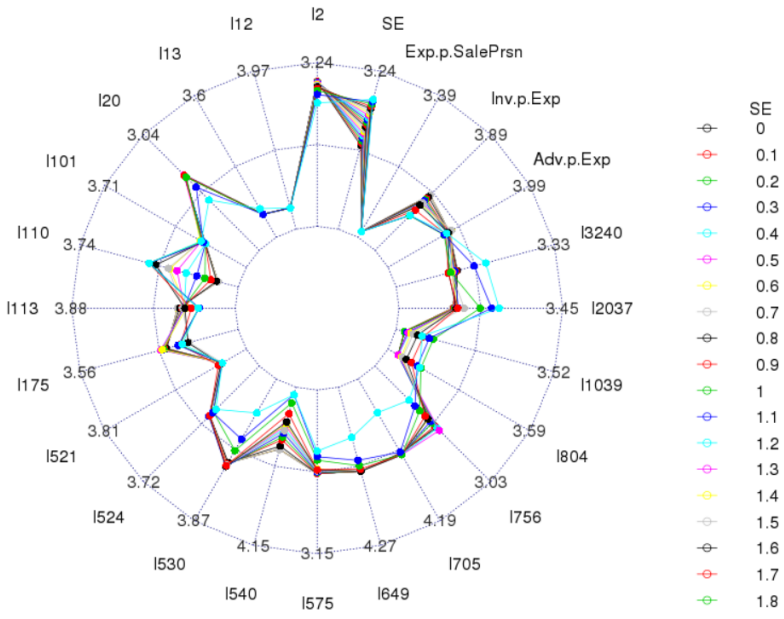


Figure 5.8: KPIs optimal values for diggerent values of $\tilde{SE}$

a profitability/SE that is far from their current performance. To overcome this issue, we change the formulation and bound the profit (P in equation (5.3)) as well as SE, so that we can control the increase in profit that can be achieved by analyzing the abilities of a particular dealer. The new formulation is displayed in equation (5.4)

Figure 5.7 shows how we can set goals for both SE and P for generating recommendations based on specific needs of dealers. Note that this plot is generated for the case that both $\tilde{SE}$ and $\tilde{P}$ are set to be equal. Obviously, the choice is not limited to that and we can set any bounds for (SE,P) pair to control in what directions (and by how much) a dealer

wishes (needs) to improve.

$$\max \quad t \tag{5.4a}$$

$$\text{s.t.} \quad P \leq \tilde{P} \tag{5.4b}$$

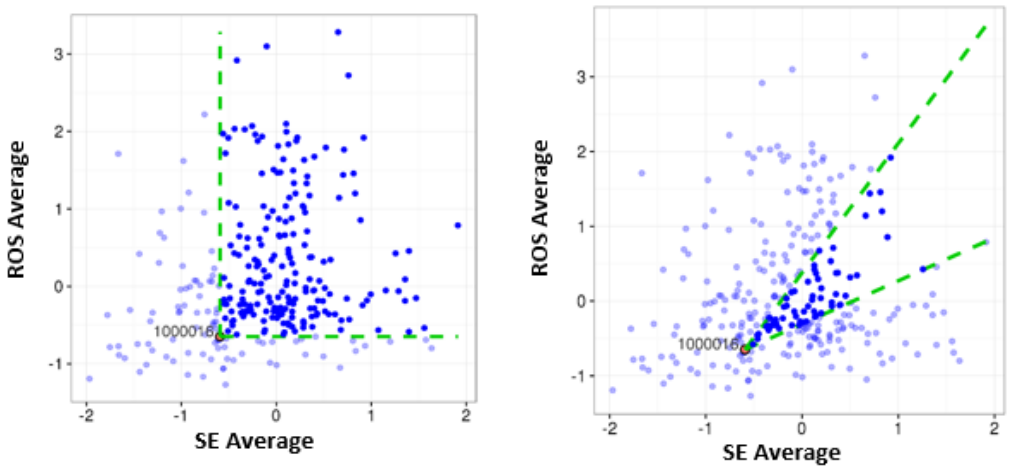
$$SE \geq \tilde{SE} \tag{5.4c}$$

$$\left|P - (\beta 0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p)\right| \leq k\hat{\sigma}_{\epsilon_P} \tag{5.4d}$$

$$\left|SE - (\gamma_0 + \gamma_1 x_1 + \gamma_2 x_2 + \cdots + \gamma_p x_p)\right| \leq k\hat{\sigma}_{\epsilon_{SE}} \tag{5.4e}$$

$$\left|x_i - (\alpha_{i0} + \alpha_{i1} x_1 + \alpha_{i2} x_2 + \cdots + \alpha_{ip} x_p)\right| \leq k\hat{\sigma}_{\epsilon_{x_i}} \quad \alpha_{ii} = 0 \quad \forall i \tag{5.4f}$$

Another issue is that the current regression models that account for objective functions constraints as well as constraints are found using the data from all the dealers (in a cluster). This approach may be problematic in that better dealers are being averaged with weaker ones to find the regression models and generate recommendations. Therefore, it is better to define which dealers should be grouped together (in terms of their performance on both SE and profitability) to develop the regression models and generate recommendations through MOO. Two possible solutions are illustrated in figure 5.9. It shows an average dealer (shown in red with its ID number), where in figure 5.9a, only the dealers whose SE and ROS average are higher in a 90° direction are selected. In this case, only the data from these dealers are used to construct the regression models (used by MOO). This approach ensures that a dealer will not be mixed with other dealers who on average are weaker than him in performance. Figure 5.9b is another possible way to define and select ”better dealers” by using a 30° cone. It is obvious that the choice of defining better dealers is not



(a) (b)

Figure 5.9: Modeling with better dealers: (a) 90° solution (b) 30° solution

limited to the above, and can be customized by the needs recognized by domain experts.

5.3 Conclusion

Increasing availability of data combined with improvements in computational platforms and technology is enabling more comprehensive and in-depth data analysis in the world of business. In the retail sector, individual stores need to utilize the available data to improve both their efficiency and effectiveness for survival and dominance. We propose an inclusive data-driven analytics platform for benchmarking and optimizing retail store performance. The proposed methodology segments stores using model based clustering.

Tailored recommendations for individual stores are extracted from associated FMR models by solving a multi-objective optimization problem to improve the profitability while controlling other performance metrics to meet the expectations of different stakeholders.

CHAPTER 6: CONCLUSION AND DIRECTIONS FOR FUTURE RESEARCH

The main objective for this dissertation was to develop a complete data-driven platform for analyzing, processing, and modeling data from retail industry in order to understand the behavior of network of stores and provide scientific managerial guidance on how to improve and operate an individual as well as groups of stores. To achieve this goal, we addressed the problem of mixture models with group structure, and noted that this has not been addressed in the literature with the existence of a dependent variable (mixture of regressions with group structure). We propose two methods to solve this problem: Mixture Models with Competitive Learning (*MMCL*) and Group Mixture of Regression (GMR) models.

MMCL is an iterative, heuristic algorithm based on Competitive Learning to cluster groups of observations and provide a model for each group. It is a non parametric approach that can be combined with any underlying regression modeling technique. We introduced an extension to *MMCL* called *MMCL++* to smartly select the initial groups.

On the other hand, GMR provides a solution to this problem by employing Expectation Maximization to find the maximum likelihood estimate for the parameters of the model. The strength of this approach is its robustness and accuracy to recover true clusters (proved by running synthetic experiments), and Maximum a Posterior (MAP) prediction density that utilizes the prior group membership information to improve the accuracy of prediction.

The effectiveness of both algorithms (MMCL and GMR) are demonstrated using syn-

thetic experiments as well as the retail (automotive dealership) case study.

The other contribution of this dissertation is the development of a multi-objective optimization (MOO) formulation for deriving recommendations in retail application settings under the result of clustering formed by mixture of regressions with group structure. We also present a real-world case study from automotive industry that aims to improve the performance of a dealership network.

There are several avenues for potential future research. The current version of GMR assumes that the covariates (features) are deterministic. This assumption can be extended to develop a model that treats the covariates as random variables. It can also be extended to Generalized Linear Models (GLM) as the assumption for the relation between the independent and dependent variables.

As for *MMCL*, it can be improved to be combined with recommendation process, meaning to judge the models based on the quality of recommendations. This can be an online reinforcement learning framework that assesses the result of recommendation and utilizes the result and feedback to adjust and improve the models.

Finally, current MOO formulation can be extended to form a pure multi-objective optimization formulation that jointly optimizes both objective functions (rather than maximizing a single objective while imposing constraints on the performance of the remaining objectives).

We believe that this research is a good starting point for developing an intensive and complete process for benchmarking and managing the performance of retail stores.

APPENDIX A: EM UPDATES IN ALGORITHM 3

Expanding the expected log-likelihood (3.6) using the definition of $\gamma_{rk}(\theta)$ in (3.3), we have

$$F(\theta;\widehat{\theta}) = E_{z\sim\tau(\widehat{\theta})}[\ell(\theta;z)] = \sum_{k=1}^{K}\tau_{+k}(\widehat{\theta})\log\pi_k + \sum_{r=1}^{R}\sum_{k=1}^{K}\sum_{i=1}^{n_r}\tau_{rk}(\widehat{\theta})\log\phi_{\sigma_k}\big(y_{ri}-\beta_k^T x_{ri}\big). \tag{1}$$

where $\phi_\sigma(t) := (2\pi\sigma^2)^{-1/2}\exp(-\frac{1}{2}t^2/\sigma^2)$ is the density of $N(0,\sigma^2)$.

We would like to maximize (1) over θ. Recall that $\beta_k, x_{ri} \in \mathbb{R}^p$ where p is the number of features. We will use $\doteq_\pi$ for example, when the two sides are equal up to additive constants, as functions of π. Fixing everything and maximizing over $\pi = (\pi_1,\ldots,\pi_k)$, we are maximizing $\pi \mapsto \sum_k \tau_{+k}(\widehat{\theta})\log\pi_k$ over probability vector π. This is the MLE in the multinomial family and the solution is $\pi_k \propto_k \tau_{+k}$, that is

$$\pi_k = \frac{\tau_{+k}}{\sum_{k'}\tau_{+k'}} = \frac{\tau_{+k}}{R} \tag{2}$$

where we used $\sum_{k'}\tau_{+k'} = \sum_{k'}\sum_r \tau_{rk'} = \sum_r\sum_{k'}\tau_{rk'} = \sum_r 1 = R$, since for fixed r, τ_{rk} sums to 1 over k.

To maximize over β, we again fix everything else. Since $\log\phi_\sigma(t) \doteq_t -\frac{1}{2}(\log\sigma^2 +$

t^2/σ^2), we are maximizing

$$\begin{aligned} F(\theta;\widehat{\theta}) &\doteq_\beta -\sum_r \sum_k \sum_i^{n_r} \tau_{rk}(\widehat{\theta}) \frac{1}{2\sigma_k^2}(y_{ri} - \beta_k^T x_{ri})^2 \\ &\doteq_\beta -\sum_r \sum_k \sum_i^{n_r} \tau_{rk}(\widehat{\theta}) \frac{1}{2\sigma_k^2}[(\beta_k^T x_{ri})^2 - 2y_{ri}\beta_k^T x_{ri}] \end{aligned} \tag{3}$$

ignoring the constant terms generated by y_{ri}^2. Note that $(\beta_k^T x_{ri})^2 = (\beta_k^T x_{ri})(x_{ri}^T \beta_k) = \beta_k^T (x_{ri} x_{ri}^T)\beta_k$. Similarly, $y_{ri}\beta_k^T x_{ri} = \beta_k^T (y_{ri} x_{ri})$. Let us define

$$\widehat{\Sigma}_r := \frac{1}{n_r}\sum_{i=1}^{n_r} x_{ri}x_{ri}^T, \quad \widehat{\rho}_r := \frac{1}{n_r}\sum_{i=1}^{n_r} y_{ri}x_{ri} \tag{4}$$

Summing over i first in (3), we get

$$\begin{aligned} F(\theta;\widehat{\theta}) &\doteq_\beta -\sum_r \sum_k \frac{\tau_{rk}}{2\sigma_k^2} n_r [\beta_k^T \widehat{\Sigma}_r \beta_k - 2\beta_k^T \widehat{\rho}_r] \\ &= -\sum_k \frac{1}{2\sigma_k^2} \sum_r \tau_{rk} n_r [\beta_k^T \widehat{\Sigma}_r \beta_k - 2\beta_k^T \widehat{\rho}_r] \end{aligned} \tag{5}$$

Let us define $w_{rk} := n_r \tau_{rk}$ and $\breve{w}_{rk} := w_{rk}/w_{+k}$ where $w_{+k} = \sum_r n_r \tau_{rk}$, and let

$$\widetilde{\Sigma}_k := \sum_{r=1}^{R} \breve{w}_{rk}\widehat{\Sigma}_r, \quad \widetilde{\rho}_k := \sum_{r=1}^{R} \breve{w}_{rk}\widehat{\rho}_r. \tag{6}$$

Dividing and multiplying by w_{+k} and summing over r in (5), we get $F(\theta;\widehat{\theta}) \doteq_\beta -\sum_k \frac{w_{+k}}{2\sigma_k^2}[\beta_k^T \widetilde{\Sigma}_k \beta_k - 2\beta_k^T \widetilde{\rho}_k]$. The problem is separable in k, and the minimizer over β_k is $\beta_k = \widetilde{\Sigma}_k^{-1}\widetilde{\rho}_k$.

To optimize over $\alpha_k := \sigma_k^2$, let us fix everything else. We have

$$F(\theta;\widehat{\theta}) \doteq_\alpha -\frac{1}{2}\sum_k \Big[\sum_r \sum_i^{n_r} \tau_{rk}\log\alpha_k + \sum_r\sum_i^{n_r}\tau_{rk}\frac{(y_{ri}-\beta_k^T x_{ri})^2}{\alpha_k}\Big]. \tag{7}$$

The first term in brackets is $(\sum_r n_r\tau_{rk})\log\alpha_k = w_{+k}\log\alpha_k$. Defining

$$E_{rk} := E_{rk}(\beta) := \frac{1}{n_r}\sum_i^{n_r}(y_{ri}-\beta_k^T x_{ri})^2, \qquad \overline{E}_k := \overline{E}_k(\beta) := \sum_r \check{w}_{rk}E_{rk}. \tag{8}$$

we see that the second term in brackets in (7) is just $w_{+k}\overline{E}_k$. We have

$$F(\theta;\widehat{\theta}) \doteq_\alpha -\frac{1}{2}\sum_k w_{+k}\Big[\log\alpha_k + \frac{\overline{E}_k}{\alpha_k}\Big] \tag{9}$$

This problem is separable in α_k and the solution is $\alpha_k = \overline{E}_k$. Putting the pieces together, we obtain the Algorithm 3.

APPENDIX B: DETAILS OF CALCULATING β ERROR

Let $\widehat{C}_k \subset [R]$ be the kth estimated cluster (the set containing indices of the groups estimated to be in cluster k) and $\widehat{z}_r \in \{0,1\}^K$ the estimated membership vector for group r, so that $\widehat{z}_{rk} = 1\{r \in \widehat{C}_k\}$. Similarly, let $C_k \subset [R]$ be the true cluster k and z_r the true label vector for group r, so that $z_{rk} = 1\{z_r \in C_k\}$. The normalized confusion matrix $F = (F_{k\ell}) \in [0,1]^{K\times K}$ between the two sets of labels is given by $F_{k\ell} = \frac{1}{R}\sum_{r=1}^{R} z_{rk}\ \widehat{z}_{r\ell} = \frac{1}{R}\sum_{r=1} 1\{r \in C_k,\, r \in \widehat{C}_\ell\}$. We have

$$
\begin{aligned}
\frac{1}{R}\sum_{r=1}^{R} \|\widehat{\beta}^{(r)} - \beta^{(r)}\|^2 &= \frac{1}{R}\sum_{r=1}^{R} \Big[\sum_{k,\ell=1}^{K} 1\{r \in C_k,\, r \in \widehat{C}_\ell\}\Big] \|\widehat{\beta}^{(r)} - \beta^{(r)}\|^2 \\
&= \sum_{k,\ell=1}^{K} \frac{1}{R}\sum_{r=1}^{R} 1\{r \in C_k,\, r \in \widehat{C}_\ell\} \|\widehat{\beta}^{(r)} - \beta^{(r)}\|^2 \\
&= \sum_{k,\ell=1}^{K} \frac{1}{R}\sum_{r=1}^{R} 1\{r \in C_k,\, r \in \widehat{C}_\ell\} \|\widehat{\beta}_\ell - \beta_k\|^2 \\
&= \sum_{k,\ell=1}^{K} \|\widehat{\beta}_\ell - \beta_k\|^2 \frac{1}{R}\sum_{r=1}^{R} 1\{r \in C_k,\, r \in \widehat{C}_\ell\} \\
&= \sum_{k,r} D_{kr} F_{kr} = \mathrm{tr}(D^T F)
\end{aligned}
$$

as desired.

REFERENCES

Definition of 'retail analytics'. https://www.collinsdictionary.com/us/dictionary/english/retail-analytics, 2017.

Rick L Andrews and Imran S Currim. Retention of latent segments in regression-based marketing models. *International Journal of Research in Marketing*, 20(4):315–321, 2003.

Rick L. Andrews, Imran S. Currim, and Peter S. H. Leeflang. A comparison of sales response predictions from demand models applied to store-level versus panel data. *Journal of Business and Economic Statistics*, 29(2):319–326, 2011. ISSN 07350015. URL http://www.jstor.org/stable/25800803.

David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

PV Balakrishnan, Anand Desai, and James Edward Storbeck. Efficiency evaluation of retail outlet networks. *Environment and Planning B: Planning and Design*, 21(4):477–488, 1994.

Yaakov Bar-Shalom. Tracking methods in a multitarget environment. *Automatic Control, IEEE Transactions on*, 23(4):618–626, 1978.

Sugato Basu. *Constrained clustering : Advances in algorithms, theory, and applications*. CRC Press, Boca Raton, 2009. ISBN 9781584889960.

Michael Bierbrauer, Stefan Trück, and Rafał Weron. Modeling electricity prices with regime switching models. In *Computational Science-ICCS 2004*, pages 859–867. Springer, 2004.

Stefano Biondi, Armando Calabrese, Guendalina Capece, Roberta Costa, and Francesca Di Pillo. A new approach for assessing dealership performance: An application for the automotive industry. *International Journal of Engineering Business Management*, 5:18, 2013.

Dankmar Böhning. *Computer-assisted analysis of mixtures and applications: meta-analysis, disease mapping and others*, volume 81. CRC press, 2000.

Randolph E Bucklin and Sunil Gupta. Commercial use of upc scanner data: Industry and academic perspectives. *Marketing Science*, 18(3):247–273, 1999.

Douglas W Caves, Laurits R Christensen, and W Erwin Diewert. The economic theory of index numbers and the measurement of input, output, and productivity. *Econometrica: Journal of the Econometric Society*, pages 1393–1414, 1982.

Gilles Celeux and Jean Diebolt. The sem algorithm: a probabilistic teacher algorithm derived from the em algorithm for the mixture problem. *Computational statistics quarterly*, 2(1):73–82, 1985.

Gilles Celeux and Gérard Govaert. A classification em algorithm for clustering and two stochastic versions. *Computational statistics & Data analysis*, 14(3):315–332, 1992.

Arun Tejasvi Chaganty and Percy Liang. Spectral experts for estimating mixtures of linear regressions. *arXiv preprint arXiv:1306.3729*, 2013.

Ashok Chaurasia and Ofer Harel. Using aic in multiple linear regression framework with multiply imputed data. *Health Services and Outcomes Research Methodology*, 12(2-3): 219–233, 2012.

Adam Cooper et al. What is analytics? definition and essential characteristics. *CETIS*

Analytics Series, 1(5):1–10, 2012.

C. Samuel Craig, Avijit Ghosh, and Sara McLafferty. Models of the retail location process: A review. *Journal of Retailing*, 60(1):5, 1984. ISSN 00224359.

Richard D De Veaux. Mixtures of linear regressions. *Computational Statistics & Data Analysis*, 8(3):227–245, 1989.

Kalyanmoy Deb. *Multi-objective optimization using evolutionary algorithms*, volume 16. John Wiley & Sons, 2001.

Kaushik Deb and Himanshu Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints. *Evolutionary Computation, IEEE Transactions on*, 18(4):577–601, 2014.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

Jean Diebolt and Christian P Robert. Estimation of finite mixture distributions through bayesian sampling. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 363–375, 1994.

Naveen Donthu and Boonghee Yoo. Retail productivity assessment using data envelopment analysis. *Journal of retailing*, 74(1):89–105, 1998.

Susana Faria and Gilda Soromenho. Fitting mixtures of linear regressions. *Journal of Statistical Computation and Simulation*, 80(2):201–225, 2010.

Cyril Goutte, Peter Toft, Egill Rostrup, Finn Å Nielsen, and Lars Kai Hansen. On clustering fmri time series. *NeuroImage*, 9(3):298–310, 1999.

Cyril Goutte, Lars Kai Hansen, Matthew G Liptrot, and Egill Rostrup. Feature-space clustering for fmri meta-analysis. *Human brain mapping*, 13(3):165–183, 2001.

Paul E. Green, Frank J. Carmone, and David P. Wachspress. Consumer segmentation via latent class analysis. *Journal of Consumer Research*, 3(3):170–174, 1976. ISSN 00935301, 15375277. URL http://www.jstor.org/stable/2488902.

Bettina Grün and Friedrich Leisch. Finite mixtures of generalized linear regression models. *Recent advances in linear models and related areas*, pages 205–230, 2008.

Sudipto Guha and Nina Mishra. Clustering data streams. In *Data Stream Management*, pages 169–187. Springer, 2016.

Sachin Gupta and Pradeep K. Chintagunta. On using demographic variables to determine segment membership in logit mixture models. *Journal of Marketing Research*, 31(1): 128–136, 1994. ISSN 00222437. URL http://www.jstor.org/stable/3151952.

Krystina Gustafson. Retail bankruptcies march toward post-recession high. http://www.cnbc.com/2017/03/31/retail-bankruptcies-march-toward-post-recession-high.html, 2017.

Lauren A Hannah, David M Blei, and Warren B Powell. Dirichlet process mixtures of generalized linear models. *Journal of Machine Learning Research*, 12(Jun):1923–1953, 2011.

Trevor Hastie and Rahul Mazumder. *softImpute: Matrix Completion via Iterative Soft-*

Thresholded SVD, 2015. URL http://CRAN.R-project.org/package=softImpute. R package version 1.4.

Simon S Haykin, Simon S Haykin, Simon S Haykin, and Simon S Haykin. *Neural networks and learning machines*, volume 3. Pearson Upper Saddle River, NJ, USA:, 2009.

N. Henke, J. Bughin, M. Chui, J. Manyika, T. Saleh, B. Wiseman, and G. Sethupathy. The age of analytics: Competing in a data-driven world. http://www.mckinsey.com/business-functions/mckinsey-analytics/our-insights/the-age-of-analytics-competing-in-a-data-driven-world, 2016.

Mehrdad Honarkhah and Jef Caers. Stochastic simulation of patterns using distance-based pattern modeling. *Mathematical Geosciences*, 42(5):487–517, 2010.

Merrilee Hurn, Ana Justel, and Christian P Robert. Estimating mixtures of regressions. *Journal of Computational and Graphical Statistics*, 12(1):55–79, 2003.

Charles A Ingene and Robert F Lusch. Market selection decisions for department stores. *Journal of Retailing*, 56(3):21–40, 1980.

Arun K Jain and Vijay Mahajan. Evaluating the competitive environment in retailing using multiplicative competitive interactive models. *Research in marketing*, 2:217–235, 1979.

Kamel Jedidi, Harsharanjeet S. Jagpal, and Wayne S. DeSarbo. Finite-mixture structural equation models for response-based segmentation and unobserved heterogeneity. *Marketing Science*, 16(1):39–59, 1997. ISSN 07322399, 1526548X. URL http://www.jstor.org/stable/184129.

Thomasz L Kamakura and Brain T Ratchfrord. Productivity assessment of multiple retail outlets. *Journal of retailing*, 72(4):333–356, 1996.

Robert P King, Timothy A Park, et al. Modeling productivity in supermarket operations: incorporating the impacts of store characteristics and information technologies. *Journal of Food Distribution Research*, 35:42–55, 2004.

Vipin Kumar and Kiran Karande. The effect of retail store environment on retailer performance. *Journal of Business Research*, 49(2):167–181, 2000.

Wolfgang Lemke. *Term structure modeling and estimation in a state space framework*, volume 565. Springer Science & Business Media, 2006.

Bruce G Lindsay. Mixture models: theory, geometry and applications. In *NSF-CBMS regional conference series in probability and statistics*, pages i–163. JSTOR, 1995.

R Lletı, M Cruz Ortiz, Luis A Sarabia, and M Sagrario Sánchez. Selecting variables for k-means cluster analysis by using a genetic algorithm that optimises the silhouettes. *Analytica Chimica Acta*, 515(1):87–100, 2004.

Ghahramani M. The information criterion. *Journal of Modern Applied Statistical Methods*, 13(2):444–454, 2014.

Geoffrey McLachlan and David Peel. *Finite mixture models*. John Wiley & Sons, 2004.

Geoffrey J McLachlan and Kaye E Basford. *Mixture models: Inference and applications to clustering*, volume 84. Marcel Dekker, 1988.

Dave Nash, Doug Armstrong, and Michael Robertson. Customer experience 2.0: How data, technology, and advanced analytics are taking an integrated, seamless customer experience to the next frontier. *Medill Department of Integrated Marketing Communications*, 32, 2013.

Simon Newcomb. A generalized theory of the combination of observations so as to obtain the best result. *American Journal of Mathematics*, pages 343–366, 1886.

Leonard J Parsons. Productivity versus relative efficiency in marketing: past and future? In *Research traditions in marketing*, pages 169–200. Springer, 1994.

Gabor Pauler, Minakshi Trivedi, and Dinesh Kumar Gauri. Assessing store performance models. *European Journal of Operational Research*, 197(1):349–359, 2009.

Karl Pearson. Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London. A*, 185:71–110, 1894.

Dan Pelleg, Andrew W Moore, et al. X-means: Extending k-means with efficient estimation of the number of clusters. In *ICML*, volume 1, pages 727–734, 2000.

H Permuter, J Francos, et al. Gaussian mixture models of texture and colour for image database retrieval. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 3, pages III–569. IEEE, 2003.

Richard E Quandt and James B Ramsey. Estimating mixtures of normal distributions and switching regressions. *Journal of the American statistical Association*, 73(364):730–738, 1978.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2016. URL https://www.R-project.org/.

Andrew M Raim, Nagaraj K Neerchal, and Jorge G Morel. An extension of generalized linear models to finite mixture outcome distributions. *Journal of Computational and Graphical Statistics*, (just-accepted), 2017.

Douglas Reynolds, Richard C Rose, et al. Robust text-independent speaker identification using gaussian mixture speaker models. *Speech and Audio Processing, IEEE Transactions on*, 3(1):72–83, 1995.

Darrell Rigby. The future of shopping. *Harvard Business Review*, 89(12):65–76, 2011.

David E Rumelhart, James L McClelland, PDP Research Group, et al. *Parallel distributed processing*, volume 1. IEEE, 1988.

Martín Safe, Jessica Carballido, Ignacio Ponzoni, and Nélida Brignole. On stopping criteria for genetic algorithms. In *Brazilian Symposium on Artificial Intelligence*, pages 405–413. Springer, 2004.

Marko Sarstedt. Market segmentation with mixture regression models: Understanding measures that guide model selection. *Journal of Targeting, Measurement and Analysis for Marketing*, 16(3):228–246, 2008. ISSN 1479-1862. URL http://dx.doi.org/10.1057/jt.2008.9.

Hugo Steinhaus. Sur la division des corp materiels en parties. *Bull. Acad. Polon. Sci*, 1 (804):801, 1956.

Yannis Stylianou, Yannis Pantazis, Felipe Calderero, Pedro Larroy, Francois Severin, Sascha Schimke, Rolando Bonal, Federico Matta, and Athanasios Valsamakis. Gmm-based multimodal biometric verification. In *eNTERFACE 2005 The summer Workshop on Multimodal Interfaces July 18th–August 12th, Facultè Polytechnique de Mons, Belgium*, 2005.

Zhaohao Sun, Kenneth Strang, and Sally Firmin. Business analytics-based enterprise information systems. *Journal of Computer Information Systems*, 57(2):169–178, 2017.

Rhonda R Thomas, Richard S Barr, William L Cron, and John W Slocum. A process for evaluating retail store efficiency: a restricted dea approach. *International Journal of Research in Marketing*, 15(5):487–503, 1998.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

Michael Tuma and Reinhold Decker. Finite mixture models in market segmentation: A review and suggestions for best practices. *Electronic Journal of Business Research Methods*, 11(1):2–15, 05 2013a.

Michael Tuma and Reinhold Decker. Finite mixture models in market segmentation: a review and suggestions for best practices. *Electronic Journal of Business Research Methods*, 11(1), 2013b.

Dany Vyt. Retail network performance evaluation: a dea approach considering retailers' geomarketing. *International Review of Retail*, 18(2):235–253, 2008.

Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schrödl. Constrained k-means clustering with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 577–584, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1-55860-778-1.

Rockney G Walters and Scott B MacKenzie. A structural equations analysis of the impact of price promotions on store performance. *Journal of marketing research*, pages 51–63, 1988.

Michel Wedel and Wayne S DeSarbo. A mixture likelihood approach for generalized linear models. *Journal of Classification*, 12(1):21–55, 1995.

Michel Wedel and Wayne S. Desarbo. Market segment derivation and profiling via a finite mixture model framework. *Marketing Letters*, 13(1):17–25, 02 2002.

Rui Xu and Donald Wunsch. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678, 2005.

Wantao Yu and Ramakrishnan Ramanathan. An assessment of operational efficiency of retail firms in china. *Journal of Retailing and Consumer Services*, 16(2):109–122, 2009.

ABSTRACT

MIXTURE MODELS WITH GROUPING STRUCTIRE: RETAIL ANALYTICS APPLICATIONS

by

HAIDAR ALMOHRI

May 2018

Advisor: Dr. Ratna Babu Chinnam

Major: Industrial Engineering

Degree: Doctor of Philosophy

Growing competitiveness and increasing availability of data is generating tremendous interest in data-driven analytics across industries. In the retail sector, stores need targeted guidance to improve both the efficiency and effectiveness of individual stores based on their specific location, demographics, and environment. We propose an effective data-driven framework for internal benchmarking that can lead to targeted guidance for individual stores. In particular, we propose an objective method for segmenting stores using a model-based clustering technique that accounts for similarity in store performance dynamics. It relies on effective Finite Mixture of Regression (FMR) techniques for carrying out the model-based clustering with grouping structure ('must-link' constraints) and modeling store performance. We propose two alternate methods for FMR with grouping structure: 1) Competitive Learning (CL) and 2) Expectation Maximization (EM). The CL method can support both linear and non-linear regression methods whereas the more effective proposed

EM approach only supports linear regression.

We also propose an optimization framework to derive tailored recommendations for individual stores within store clusters that jointly improves profitability for the store while also improving sales to satisfy franchiser requirements. We validate the methods using synthetic experiments as well as a real-world automotive dealership network study for a leading global automotive manufacturer.

AUTOBIOGRAPHICAL STATEMENT

Haidar Almohri received his B.Sc. degree in Electronic Engineering Technology and M.Sc. degree in Electrical Engineering from the University of Hartford, CT.

He worked with Siemens Kuwait as an electrical engineer for five years before starting his second M.Sc. degree in Industrial and Operations Engineering at the University of Michigan, Ann Arbor, MI. He successfully completed his degree and then started his PhD in Industrial Engineering at Wayne State University, where he is currently a PhD candidate.

His research interests include Big Data, High-Dimensional Data Analysis, Business Analytics, Predictive Analytics, Mixture Models, Causal Analysis, Multi-objective Optimization.